Eee » ALSA » Rails » Sysadmin » Qt 4.4

LINUX FORMAT

Главное в мире Linux

LXFDVD

Fedora 9

Крепко скроенный Linux с 2000+ пакетов!

Еще дистрибутивы » Учебники » Игры » и так далее!

Школа LXF. Перейдем на L?
Опыт решения проблем миграции на свободное ПО в сфере образования с. 92

4 607130 821718

Август 2008 № 8 (108)

LINUX НА ЧЕМ УГОДНО

Владельцы КПК и приставок – хотите повеселиться? Мы поставим вам Linux!

Продвинутый Eee
Добавьте еще один дистрибутив на SD-карту с. 58

Чиним ALSA
Заставьте свою аудиоплату работать с. 34

Ruby on Rails
Минимум кода – максимум отдачи с. 66

«Раньше X мог запросто жечь ЭЛТ-мониторы, но сейчас они отлично защищены».

Кейт Паккард, спаситель ~~вселенной~~ рабочего стола с. 30

Каталог агентства «РОСПЕЧАТЬ» – подписной индекс 20882
Каталог «ПРЕССА РОССИИ» – подписной индекс 87974





Что мы делаем?

» Мы поддерживаем открытое сообщество, предоставляя источник информации и место для обмена мнениями.

» Мы помогаем нашим читателям узнать больше о Linux в нашей рубрике Учебники: здесь каждый найдет себе что-то по вкусу!

» Мы лицензируем весь код, который появляется на страницах журнала, на условиях GPLv3.

» Мы стараемся предоставлять максимально точные, непредвзятые и актуальные сведения о мире Linux.

Кто мы?

«Если бы у вас был большой-пребольшой Linux-кластер, то на что бы вы потратили все эти терафлопсы?» – мнение команды LXF.



Грэм Моррисон
Я бы заставил этот железный мозг думать, зачем он нужен. А сам пошел бы в пивную.



Майк Сондерс
На борьбу с мифами и суевериями – это почти как сворачивание белка, но неврологичнее. Все, что вам говорят – ложь. Я ЕСТЬ ИСТИНА!



Нейл Ботвик
Пусть прикончит все остальные системы и заставит людей втиснуться в рамки Sinclair 16K. Хватит с вас Jetpac!



Эфраин Эрнандес-Мендоса
Это позволит слегка разгрузить компьютер NASA, составляющий мой барный счет. 14 млрд., 70 млрд. – какая разница?



Эндрью Грегори
Связать весь мир через сети www.justfortheloveofit.org и, смеясь, наблюдать, как деньги превращаются в пыль.



Энди Ченнел
Попробовал бы просчитать, почему группы вроде The Kooks и The Feeling столь популярны. Я хотел сказать, какого черта...



Дэвид Картрайт
Поделился бы с пользователями OpenOffice.org, чтобы они могли работать с документами чуточку быстрее.



Майкл Дж Хэммел
Поискал бы разумную жизнь в США. Правда, чтобы отфильтровать зрителей American Idol, потребуется еще один кластер...



Джон Брэндон
Заставлю поискать ответы на вечные вопросы: «Куда уходит время?» и «Сколько дорог должен пройти человек?»



Д-р Крис Браун
На квантовую хромодинамику. Терри Пратчетт утверждает, что бесконечность синяя. Я думаю, он неправ – она коричневая.



Ник Вейч
Сгенерирую вопросы этой колонки на ближайшие 100 номеров и ответы к ним. Главное – поддеть Майка в каждом из них.



Ричард Смэдли
Декодирую музыку сфер, а потом перепишу ее с двойными басами и нотками банджо. И колокольчиком!



К вопросу о Linux на тостере

Наша любимая ОС удивительно гибка: с таким набором архитектур (уступающим разве что NetBSD) ее действительно можно запустить на чем угодно – был бы процессор и блок управления памятью, хотя, в принципе, можно обойтись и без последнего. И те гаджеты, которые попали на страницы темы номера – лишь верхушка айсберга.

Вряд ли кто-то станет спорить, что это хорошо. Запустить открытую ОС на давно валяющейся на антресоли электронной штуковине – это всегда интересно и познавательно. Профессиональные инженеры приведут в пользу Linux десяток доводов: он и бесплатен, и изменить можно все, что захочется. Красота!

Но взглянем на него с другой стороны. Я был бы рад иметь сотовый телефон, работающий под управлением Linux. И даже (может быть) писал бы для него программы. Но в первую очередь, мне нужно устройство, которое будет звонить и отправлять SMS, с дизайном, не смахивающим на Р-153.

Или возьмем фаворита – Nokia N800. У меня тоже есть такой, и я с удовольствием им пользуюсь. Но хваленый «полноценный X-сервер» на деле ворочается там не быстрее, чем KDE3 на PI. Есть и другие шероховатости. В общем, «если Вы можете засунуть горошину в ухо – это еще не повод так делать». Linux на гаджете ценен сам по себе, но за технико-идеологической стороной вопроса надо не забывать и о функционально-пользовательской. **LXF**

Валентин Синицын » Главный редактор info@linuxformat.ru

Как с нами связаться

Письма для публикации: letters@linuxformat.ru
Подписка и предыдущие номера: subscribe@linuxformat.ru
Техническая поддержка: answers@linuxformat.ru
Проблемы с дисками: disks@linuxformat.ru
Общие вопросы: info@linuxformat.ru
Web-сайт: www.linuxformat.ru
› **Адрес редакции:** Россия, Санкт-Петербург, Лиговский пр., 50, стр. 15.
› **Телефон редакции:** +7 (812) 640-49-90. Дополнительная информация на стр.120





Содержание

Весь номер – прямо как на ладони: приятного чтения!

Обзоры



› Fedora 9: не надо ждать круглой даты, чтобы попробовать!

Сравнение: менеджеры закачек



Linux на чем угодно

Установим нашу любимую ОС всюду, где есть процессор с. 22



Что за штука... T2

Что, Терминатор вернулся? Нет, Gentoo переродился! с. 42



Amazon работает на вас с. 38



Люди говорят

«KDE и Gnome мало чем отличаются: оба они – двумерные»

Кейт Паккард запускает лису в курятник с. 30





Последние новинки мира Linux – в этом мощном и универсальном дистрибутиве

- Простой и удобный инсталлятор
- Новый менеджер пакетов
- Gnome 2.22 и KDE 4.0.3
- Поддержка зашифрованных разделов
- Более 2000 приложений!



Спецрепортаж

Подпишись на Linux Format и сэкономь!

Постоянные рубрики



› Да, Linux можно запустить на каждом из этих устройств.

Учебники





LINUX FORMAT Новости

ГЛАВНЫЕ НОВОСТИ: » Linux: к школе готов » ODF – признанный формат №1 » Инфра-Ресурс и Canonical заключили соглашение » Свободу драйверам и Java! » Суперкод-2008: итоги

» Рубрику ведет Илья Шпаньков



Время разбрасывать камни

За спортивными победами и достижениями в области поп-культуры как-то незаметно и обыденно произошло довольно важное событие: успешно завершился первый этап перевода российских школ на использование свободного ПО. Не сказать, чтобы он проходил гладко: в процессе тестирования дистрибутивов в пилотных регионах было обнаружено множество мелких недочётов, но, как говорится, не ошибается только тот, кто ничего не делает. На данный момент все замечания учтены, выявленные ошибки исправлены, а наборы свободного ПО прошли аттестацию, получив соответствующие подтверждающие документы. Теперь на очереди второй этап – внедрение готовых к активному использованию свободных программных продуктов в российские школы.

Участники проекта предложили несколько дистрибутивов, ориентированных на различные области применения:

1 «Легкий Линукс» – данный вариант оптимизирован для установки и работы на компьютерах, оснащенных оперативной памятью в размере от 128 до 256 МБ и процессором от PI 233 МГц. Он обладает самыми низкими аппаратными требованиями, использует графическую среду XFCE, предоставляет пользователям лёгкий и быстрый текстовый редактор *Abiword*, электронную таблицу *Gnumeric* и другое «нетяжелое» ПО.

2 «Линукс Юниор» – более производительный дистрибутив, рассчитанный на компьютеры с ОЗУ от 256 МБ до 1 ГБ. Он базируется на графической среде KDE, а помимо офисных приложений и обучающих программ включает средства разработки, такие как *Lazarus* и *KDevelop*.

3 «Линукс Мастер» – наиболее мощная редакция: ее можно использовать как для обучения компьютерной грамотности, так и для создания полноценных программ на Java, C/C++ и PHP, благодаря входящей в состав дистрибутива среде разработки *Eclipse*. Кроме *Eclipse*, в «Линукс Мастер» входят те же программы, что включены в «Линукс Юниор».

4 «Линукс Терминал» – система, предназначенная для использования в компьютерных классах с сервером. Программы, с которыми работают школьники, фактически выполняются централизованно, а машины учеников используются только для отображения и ввода информации (так называемые «тонкие клиенты»). Благодаря такой организации работы класса аппаратные требования к машинам учеников становятся очень низкими (память от 32 МБ), а основная вычислительная нагрузка ложится на сервер.

5 «НауЛинукс» – данная версия также позволяет организовать компьютерный класс по схеме «терминальный сервер – тонкие клиенты». В этом случае основную нагрузку по выполнению приложений несет сервер, а требования к компьютерам, используемым в качестве рабочих мест учащихся, могут быть минимальными.

Столь широкий спектр доступных модификаций оказался как нельзя более кстати для российских школ, не избалованных (в основной своей массе) новинками компьютерной индустрии, поэтому ещё до окончания тестового периода организаторы проекта стали получать из регионов письма с просьбой предоставить той или иной школе возможность опробовать необычные новинки. Надо сказать, что подобная активность порадовала не только самих разработчиков, но и инициатора – Министерство информатизации и связи РФ. В результате чиновниками была подготовлена специальная форма, заполнив которую, учебные заведения могут стать участниками проекта ещё до наступления активной фазы второго этапа.

Новости короткой строкой

» Вышел финальный релиз дистрибутива OpenSUSE 11.0.

» Разработчики из компании Canonical, Ltd. подготовили специализированную редакцию дистрибутива Ubuntu 8.04 для мобильных ПК.

» Компания Xandros объявила о начале процесса приобретения другого известного разработчика настольного Linux – Linspire (ранее – Lindows).

» Подготовлен очередной стабильный релиз набора компьютерных игр, запускаемых с Live-DVD – LinuX-Gamers Live 0.9.3.

» Вышел очередной стабильный релиз дистрибутива Gentoo Linux 2008.0.



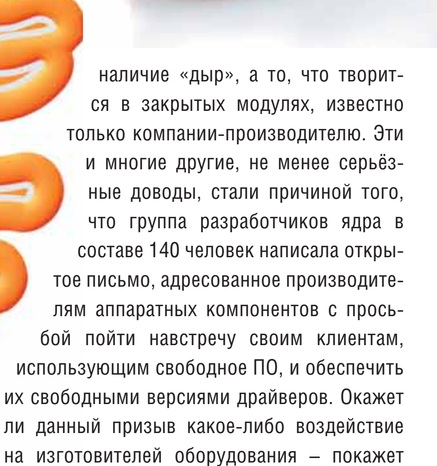
Очевидное-невероятное

Да, такого финала не предполагал никто. Оно и понятно: то, что монополист достигает своей цели во что бы то ни стало, а остальные вынуждены уж как-то подстраиваться под его изменчивые капризы, годами воспринималось нами как единственно возможный уклад жизни. Но, видимо, действительно приходят новые времена, и вот уже гигант Microsoft публично признаёт своё поражение в борьбе форматов: отныне только ODF остаётся единственным международным стандартом для офисных документов. Одновременно с объявлением об отказе от дальнейшей борьбы представители Microsoft пообещали, что полноценная поддержка ODF будет добавлена в *MS Office 2007* с пакетом обновлений SP2 в первой половине 2009 года.

Данная победа – сама по себе знаковое событие, но следует отметить, что одновременно это и своего рода отправная точка для новой эпохи рынка программного обеспечения: впервые за всю историю офисных пакетов приложений правила диктует открытый формат. Не секрет, что борьба в данном секторе рынка велась постоянно и с переменным успехом, но всегда соревнование велось между разработчиками проприетарного ПО. Теперь же все они будут обязаны играть по правилам, предложенным сторонниками совершенно иных принципов разработки и использования современных технологий. Остаётся только надеяться, что данный пример не останется в анналах истории как единичный уникальный случай, а послужит примером для всех остальных участников рынка.



Ядро без примесей



Это должно было когда-нибудь случиться. Абсолютное большинство пользователей операционных систем GNU/Linux, говоря: «Я применяю свободное ПО» были, скажем так, не совсем точны. Ведь множество дистрибутивов GNU/Linux поставляются со слегка (или не слегка) модифицированным ядром, нередко содержащим проприетарные (и, как следствие, несвободные) модули. К наиболее известным из них можно отнести оригинальные видео-драйверы от ATI или NVIDIA, драйверы модемов, беспроводных карт и многое другое. Естественно, параллельно ведутся разработки свободных аналогов (и они часто используются по умолчанию), но при этом их функциональность иногда оставляет желать лучшего: производители аппаратных компонентов не спешат открывать спецификации на свои продукты, а без этого сделать полноценный драйвер практически невозможно.

Но такая ситуация с включением проприетарного кода в ядро Linux приводит к определённым проблемам, и дело тут не только в идеологии свободного ПО. Наличие закрытых компонентов уменьшает возможности для осуществления полного цикла внесения изменений или поправок в ядро, вынуждая разработчиков подстраиваться под особенности несвободных модулей. В этом случае также снижается и защищённость ядра: свой код разработчики могут проверить на наличие «дыр», а то, что творится в закрытых модулях, известно только компании-производителю. Эти и многие другие, не менее серьёзные доводы, стали причиной того, что группа разработчиков ядра в составе 140 человек написала открытое письмо, адресованное производителям аппаратных компонентов с просьбой пойти навстречу своим клиентам, использующим свободное ПО, и обеспечить их свободными версиями драйверов. Окажет ли данный призыв какое-либо воздействие на изготовителей оборудования – покажет время.



Свобода на все 100%



Перевод на другую лицензию даже небольшого программного продукта всегда связан с ворохом проблем, а уж для массивных проектов первой величины подобная процедура может стать довольно затратной как по времени, так и по средствам. Причина в том, что за многие годы своего развития код обрастает различными дополнениями и новыми компонентами, права на которые могут принадлежать совсем не основным разработчикам. Примерно такая ситуация сложилась и с переводом Java под действие лицензии GPL: постепенный переход на свободные рельсы, начатый ещё в мае 2006 года, только сейчас завершился полным успехом – отныне Java может считаться свободной разработкой на все сто процентов.

На последнем этапе немалую помощь компании Sun Microsystems оказали разработчики Red Hat, взявшие на себя задачу по замене не подпадающего под свободную лицензию кода проекта IcedTea. В добавление к тысячам строк, переписанных заново, программисты смогли также продемонстрировать и преимущества свободного ПО при решении сложных задач: для упрощения и ускорения работы был использован код GNU Classpath, изначально лицензированный под GPL. После небольшой переделки и адаптации к уже существующему коду IcedTea, данная разработка позволила полностью восстановить функциональность последнего несвободного компонента Java, но уже в виде свободного проекта. Это подтвердили и тщательные тесты, проведённые сотрудниками Sun Microsystems и Red Hat с помощью пакета Java Test Compatibility.



И Офис превращается в...



Российский рынок ПО по-своему уникален: здесь сочетается хорошая техническая подкованность пользователей и обширная ниша для распространения новых программных продуктов. Поэтому не удивительно, что многие крупные разработчики не прочь «застолбить» для себя его определённую часть. Очередное крупное событие подобного плана произошло в июне: компания Инфра-Ресурс, давно и прочно занявшая лидирующие позиции в подготовке адаптированной для российских реалий версии пакета офисных приложений *OpenOffice.org*, заключила партнёрское соглашение с уже известной на весь мир благодаря дистрибутиву Ubuntu фирмой Canonical, Ltd. Данное соглашение позволяет Инфра-Ресурсу официально заниматься распространением самого популярного в мире дистрибутива GNU/Linux, а также осуществлять его установку, модернизацию и дальнейшее обслуживание на рынке корпоративных решений и серверных систем.

Событие носит, безусловно, положительный характер: ведь не секрет, что корпоративный сектор с опаской относится к предложениям использовать ПО от неизвестно какой «забугорной» компании. А когда есть вполне реальное юридическое лицо, с которым можно и бумаги официальные подписать, и договориться о скидках на услуги, да и просто есть кому предъявить претензии, если что-то пойдет не так, то многие вопросы отпадают сами собой и становится уже не важно, где и кем собирался тот или иной пакет, входящий в состав предлагаемого дистрибутива. Надо отметить, что компания Инфра-Ресурс решила, не откладывая в долгий ящик, заявить о себе как о новом игроке уже рынка операционных систем самым прямым образом: всего через несколько дней после обнародования информации о подписанном с создателями Ubuntu соглашении, российские разработчики представили свой вариант основанной на Ubuntu операционной системы – InfraLinux. Естественно, одной из ее жемчужин стала специализированная сборка пакета *OpenOffice.org*, подготовленная авторами нового, теперь уже российского дистрибутива.

Триумф юных дарований

В конце июня завершился уникальный конкурс Суперкод-2008, продолжавшийся в течение полугода. Принять участие в нём могли молодые люди со всех уголков России. Данное мероприятие было организовано творческой группой «ТЫРНЕТ – детский интернет» при поддержке корпорации Intel и Комитета по молодёжной политике и взаимодействию с общественными организациями г. Санкт-Петербурга. В творческом соревновании по разработке компьютерных игр и мультимедийных проектов в среде Скретч [Scratch] (LXF105) принимали участие юные программисты в возрасте от 7 до 16 лет, при этом, благодаря организации дистанционного участия в конкурсе, свои силы смогли попробовать даже жители регионов Дальнего Севера.



Изначально в отборочных турах стартовало около 500 участников, а к региональным полуфиналам, проходившим в Нижнем Новгороде, Новосибирске, Москве и Санкт-Петербурге, было допущено более 80 молодых дарований. В финал прошёл 21 участник, и в ходе упорной борьбы были выявлены следующие победители:

» I место – **Юля Юкова** (13 лет, г. Саров Нижегородской обл., школа № 5), за игру «Экскурсия по Петербургу»;

» II место – **Евгений Шибашов** (14 лет, г. Нижний Новгород, школа № 32), за игру «Убери лишние предметы»;

» III место – **Василий Голубцов** (14 лет, МОУ Иванищенской СОШ, Ярославская обл.), за игру «Прополка овощей».

ТЫРНЕТ

Кроме того, ценными призами были отмечены и работы других участников, не набравших максимальный балл, но представивших весьма оригинальные проекты. Поздравляем лауреатов и желаем им и в будущем одерживать столь же замечательные победы! LXF

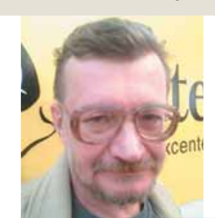
» При подготовке данного выпуска новостей были использованы материалы сайтов linux.armd.ru, www.osnews.com, www.i-rs.ru, www.linuxfoundation.org, www.press.redhat.com и www.supercode.ru.

LINUX FORMAT

Обзоры



Новинки программного и аппаратного обеспечения в описании наших экспертов



Алексей Федорчук
Его слабости – mass storage, разметка диска и файловые системы.

Какой Linux изучать?

Последнее время много говорят о внедрении Linux в школах, вузах, на производстве, забывая, что это требует знания предмета, с чем пока наблюдается напряженка. Не вырастила страна достаточного количества специалистов – их самих еще предстоит обучить. Возникает вопрос – а какому Linux? На этот счет существует две точки зрения:

1 Будущие гуру должны учиться на наиболее дружественных пользователю системах, таких как SUSE, Ubuntu, Mandriva, ALT Linux – тем более, что два последних начинают утверждаться в российских школах. Недостаток данного подхода – сложность «докапывания» до сути вещей, скрытых за «дружественными» надстройками.

2 Надо использовать «метод большого болота» – системы типа Slackware, Gentoo, Arch. Оно и ясно: выбравшемуся из него лужи малые уже нипочем. Но многие ли смогут, без предварительной подготовки?

И потому выскажу третью точку зрения: изучать надо систему, обеспечивающую поэтапное погружение, где сочетаются легкость развертывания и возможность глубоко вникнуть в суть происходящих процессов. И как минимум один дистрибутив, отвечающий этим условиям, есть: Zenwalk Linux. Развернутое доказательство автор надеется дать здесь: http://citkit.ru/articles/892/.

А уж что будет потом внедряться новообученными гуру – не имеет никакого значения.

alv@posix.ru

Сегодня мы рассматриваем...

Fedora 9 с. 08



› Если после неудачного выступления Ubuntu в LXF107 вы слегка приуныли – попробуйте Fedora.

Brain Train с. 10



› Разберитесь как работают мозги Майка, не покидая привычный рабочий стол.

НАШ ВЕРДИКТ: пояснение

Все попавшие в обзор продукты оцениваются по одиннадцатибальной шкале (10 – высшая оценка, 0 – низшая). Как правило, мы оцениваем функциональность, производительность, простоту использования и цену, а для бесплатных программ учитывается документация. Кроме того, мы всегда выставляем общую оценку, демонстрирующую наше отношение к продукту.

Выдающиеся решения могут получить престижную награду «Top Stuff». Номинантами становятся лучшие из лучших – просто высокой оценки здесь недостаточно.

LINUX FORMAT TOP STUFF

Рассматривая свободное ПО, мы обычно указываем предпочтительный дистрибутив. Иногда это означает компиляцию из исходных текстов, но, если разработчики рекомендуют Autopackage, мы следуем этому совету.



Google Earth
Разработчик: Google
Сайт: http://earth.google.com
Цена: Бесплатно по закрытой лицензии

Функциональность	10/10
Производительность	9/10
Простота использования	9/10
Документация	9/10

» *Если весь мир – сцена, то Google Earth – театр. Простая в использовании, захватывающая и ободряющие практичная программа.*

Рейтинг 9/10

Fedora 9

Fedora 9 в очередной раз задает тон в разработке свободного ПО. **Ник Вейч** остался доволен…

Вкратце...

» Fedora – спонсируемый Red Hat дистрибутив Linux общего назначения. Выпускается с 6-месячным циклом и часто служит испытательным полигоном для ПО, которое затем переходит во «взрослую», коммерческую версию Red Hat. См. также SUSE, Ubuntu и Mandriva 2008.

Ключевое ПО

- » **Ядро 2.6.25**
- » **Gnome 2.22**
- » **KDE 4.0.3**
- » **ext4**
- » ***Firefox 3 (beta)***
- » ***Gimp 2.4***

Не каждый день увидишь новую версию Fedora. В среднем она выходит каждые 182 дня, но раз в полгода мы вправе ожидать чего-нибудь этакого. И верно: налицо буквально россыпь свежего ПО и потрясающий новый рабочий стол. И, конечно, масса внутренних реформ, в том числе и долгожданных.

Если Fedora 8 показал, что команда способна выдать солидную, элегантную настольную систему, то версия 9 пошла дальше, и теперь к надежной основе добавлено новейшее ПО, в уверенности, что оно будет работать. В разумных пределах.

Что вы видите

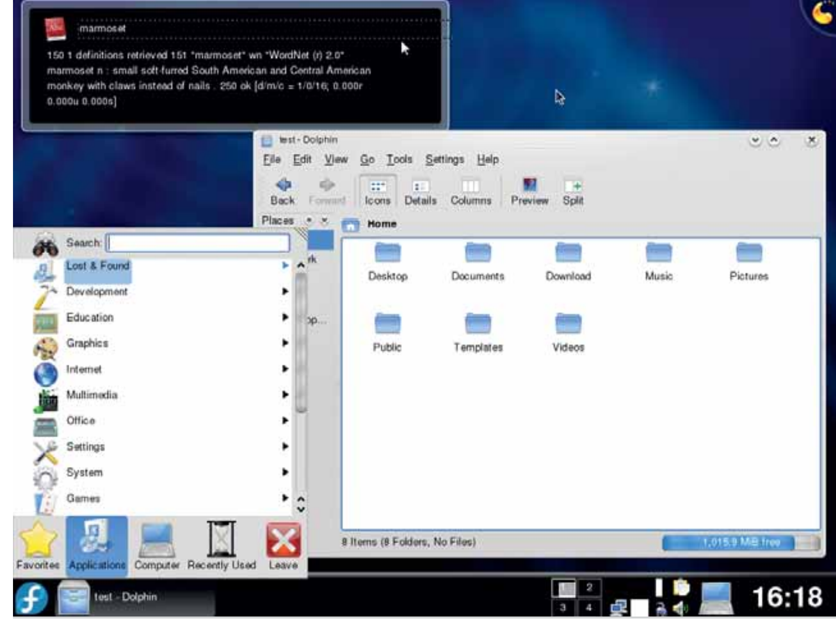
Бесспорный хит сезона – KDE 4.0.3. Хотя Gnome по-прежнему является (а может, и всегда будет) рабочим столом по умолчанию, команда Fedora легла костьми, чтобы интегрировать также и KDE 4. Причем это не пробное дополнение, а полноценный рабочий стол. Одна из проблем с KDE 4 в настоящий момент – недостаток приложений; не все, даже стандартные, KDE-программы перешли на новые библиотеки. Поэтому библиотеки KDE 3 тоже на месте, и рабочий стол фактически находится в смешанном, переходном режиме. Пользователь вряд ли это заметит, но от машины требуется скорее больше ресурсов, чем меньше (вопреки тому, за что боролись). Но тут уж Fedora ни при чем.

Новый интерфейс KDE несколько лаконичен для новичка; файловый менеджер *Dolphin* требует привычки, а новое стартовое меню, задуманное как дружелюбное, по мнению многих (включая **LXF**) стало неуклюжим. На нашем диске прилагается видео о KDE 3 и Fedora.

Невольно забываешь, что Gnome тоже улучшился (помимо очевидной смены декораций в тон с Fedora). Gnome предпочитает тактику постепенного, но неуклонного обновления. И в этом есть свои плюсы.

Долгожданный капремонт получил *Yum* и его графический интерфейс. Нет больше *pup* и *pirut*, встречайте *PackageKit*! С виду перемены не так уж радикальны, но новый менеджер работает быстрее и чище, хотя и не без про-

> «Наряду с Gnome, Fedora интегрировал рабочий стол KDE 4.»



› **Включение KDE 4 – это здорово. Интеграция замечательная, хотя и не без проблем. Похоже, в ближайшем будущем нас ждет много-много обновлений.**

Шаг за шагом: добавьте плазмоиды KDE 4!



» Отделка интерьера

В исходном состоянии KDE в Fedora имеет весьма бледный вид. Даже минималист найдет его скудным. Но все в ваших руках! Кликните на иконку палитры в правом нижнем углу.



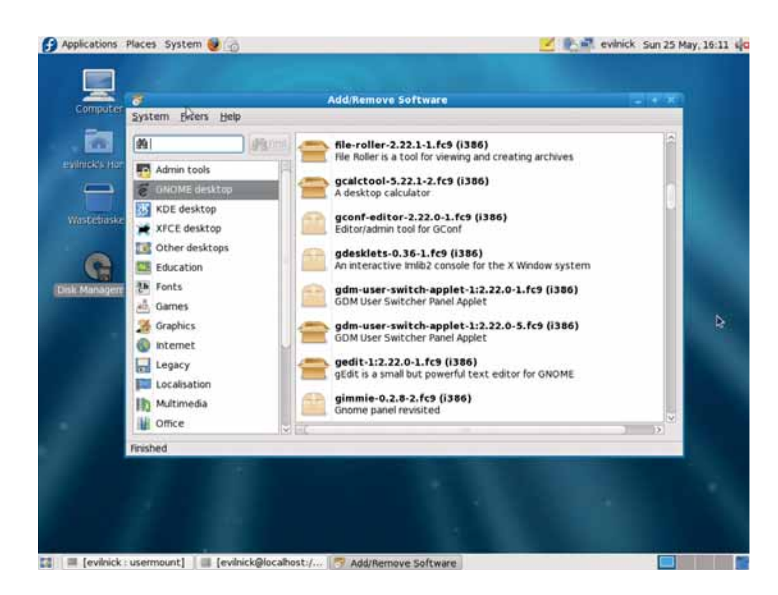
» Уж не Мас ли это?

Как по волшебству, появится диалог с перечнем доступных виджетов (это которые плазмоиды). Выберите что понравится, нажмите кнопку Add widget, и они к вашим услугам!

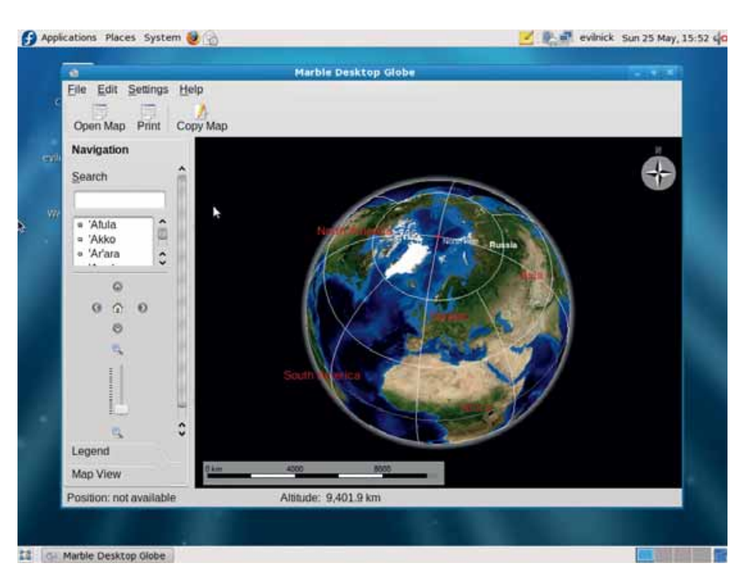


» Подкрутим гайки

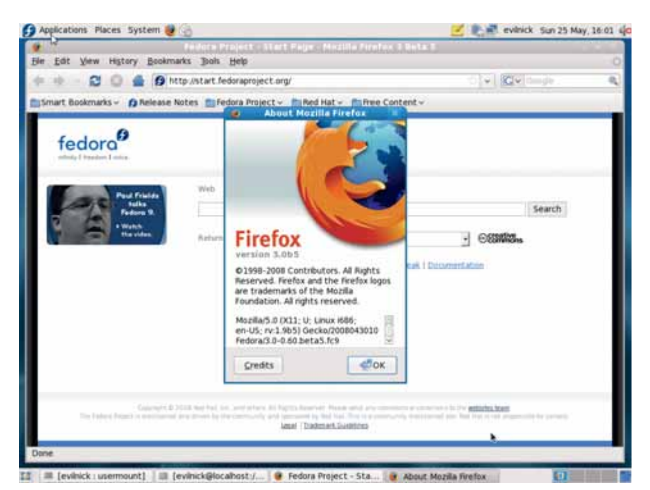
У большинства плазмоидов есть регулируемые настройки. Наведите мышь на виджет; появится окно. Щелкнув по крошечной иконке – гаечному ключу, получим диалог установки параметров.



› Управление пакетами сильно улучшилось. Интерфейс *PackageKit* куда краше предшественников и, по ощущениям, быстрее.



› Весь мир в ваших руках – пусть даже только на экране – благодаря чудесной *Marble*.



› *Firefox 3* все еще бета, поэтому кое-кто предпочтет версию 2.x.

блем (в основном связанных с ключами GPG, но это обещают починить). А боги управления пакетами вняли мольбам тех, кто работает с *yum* в терминале: теперь функция поиска выдает однострочные результаты, а не извергает полное описание всего сущего!

Чего вы не видите

Файловую систему видят разве что магнитные головки жестких дисков. Но если бы мы сами влезли вовнутрь, то увидели бы, что Fedora поддерживает ext4, и поняли бы, что это хорошо. На разработку файловых систем брошено немало программистов Red Hat, и подобной новинки следовало ожидать – ext4 очень неплоха. Что она даст лично вам? Во-первых, ext4 умеет управлять разделами размером до одного эксабайта – миллиона гигабайт! Затем, и файлы в этой системе могут быть не меньше. Как ни удивительно, ext4 эффективнее прежних систем, даром что отмечает время с точностью до наносекунд.

Что вы хотите видеть

Еще одна важная невидимка – OpenJDK6. Это полноценный JDK (Java Developer Kit), распространяемый на условиях GPLv2. Он дает больше свободы Java-приложениям, работающим на Fedora, и устраняет многие проблемы, связанные с прежними «свободными» реализациями Java. Несовместимость тут неизбежна, но за время испытаний в обычном режиме мы ничего такого не заметили.

Среди вещей, не попавших в новую версию – *Presto*, готовая только на 82 процента, судя по сообщениям на wiki проекта. *Presto* – это Fedora-вариант *Delta RPMs*. Идея новшества в том, что при обновлении пакетов далеко не всегда изменяются все файлы – например, присылается заплатка на исполняемый код, а справки это не касается. Иногда в пакет просто добавляется новый язык – такое случается сплошь и рядом! Из-за подобных мелочей менеджер обновлений принимается заново грузить огромные пакеты. *Delta RPMs* (сократители уже наготове, вот-вот окрестят это DRPM) будет загружать только то, что изменилось. Мы слышали, что технология DRPM войдет в очередную версию SUSE, и с нетерпением ожидаем ее в Fedora 10.

Итого: крепкий релиз, со многими тонкими обновлениями и новейшими версиями ПО, включая *GIMP* и *OpenOffice.org*. Ошибки, конечно, найдутся, но держу пари: зачистят их моментально. LXF

> «Файловый менеджер Dolphin потребует некоторой привычки.»

Получите Live!

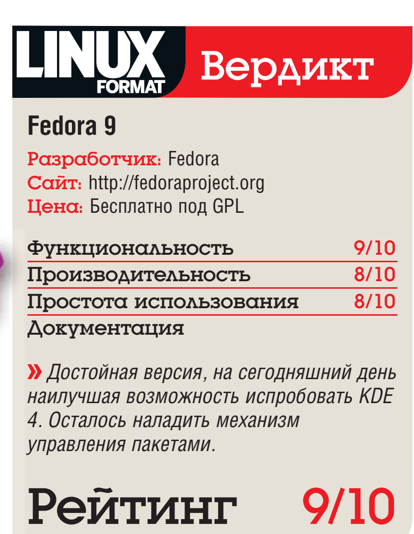


Конечно же, новая версия Fedora получит целый букет вариантов инсталляции. Хорошая новость: по приложении минимума умственных усилий, «живые» дистрибутивы на USB-брелке выучились запоминать настройки и файлы на диске.

LINUX FORMAT Вердикт

Fedora 9

Разработчик: Fedora
Сайт: http://fedoraproject.org
Цена: Бесплатно под GPL



Функциональность	9/10
Производительность	8/10
Простота использования	8/10
Документация	

» *Достойная версия, на сегодняшний день наилучшая возможность испробовать KDE 4. Осталось наладить механизм управления пакетами.*

Рейтинг 9/10

The Amazing Brain Train

Майк Сондерс изрядно настрадался от опозданий поездов. Может быть, этот забавный вояж поднимет ему настроение?

Вкратце...

» Подборка игр-головоломок по мотивам *Big Brain Academy* для Nintendo DS и Wii, но нацеленная в основном на юных игроков.

Ключевое ПО

- **» Процессор 500 МГц**
- **» 256 МБ ОЗУ**
- **» 4 МБ свободного места на диске**

Нам нравятся Grubby Games (в смысле, а компания, а не пыльные диски и картриджи). В LXF99 мы выставили игре *Professor Fizzwizzle and the Molten Mystery* рейтинг 8, оценив сочную 2D-графику и богатую коллекцию уровней (210). Теперь *Grubby* обратилась к жанру развития умственных способностей, который в свое время имел оглушительный успех на Nintendo DS – и команде из Ванкувера это прекрасно удалось!

The Amazing Brain Train (*TABT*, Удивительно Мозговитый Поезд; кроме того 'train' переводится еще и как «тренировать») – комплект из 15 малых игр-головоломок (паззлов), предназначенных для тренировки серого вещества по пяти разным категориям: арифметика, память, поиск, пространственное воображение и планирование. Не все игры доступны с самого начала – некоторые нужно «распечатать», выполнив задание. Уже видя экран-заставку, вы влюбляетесь в игру: в ее внешнюю привлекательность, мягкие фоновые рисунки и милых зверюшек, которые все вместе создают теплую благожелательную атмосферу.



› **В Birthday Cake Shuffle надо так передвинуть квадратики со зверями, чтобы Профессор добрался до праздничного пирога.**

Основной режим *TABT* – это Quest. Игра начинается с северной стороны карты и проходит в форме путешествия на поезде: в нем вы колесите по железным дорогам, встречая разных животных и разгадывая их загадки. Вы познакомитесь с черепахой, овцой, пингвином, пандой и множеством других созданий; они то предлагают вам новую головоломку, то просто останавливаются поболтать.

При выполнении некоторых заданий игра открывает новые участки железной дороги к центру карты, поэтому интерес постоянно подпитывается. Можно сохранить результат игры, и ваши достижения не пропадут. Хотя 15 игр – не так уж много, но все они очень разные, поэтому игра долго не надоедает.

Играем без устали

Заядлые игроки в пазлы, наверно, уже встречали некоторые из игр *TABT*, в том или ином виде. Математические загадки вполне предсказуемы (нужно угадать недостающее число в простом уравнении, подобрать величины, чтобы вышла сумма, и т.п.), а по части тренировки памяти предлагается запомнить изменения в группах объектов или проследить последовательность линий.

И все это в *TABT* сделано душевно. Вот, например, старая игра, в которой надо установить и повернуть должным образом зеркала, направляя солнечный луч. В *TABT* вы делаете это, чтобы согреть мишек коала (а когда это удается, они радостно пляшут для вас в благодарность). Или вспомните игру, где нужно распутать линии, потянув их за кончики. В *TABT* это спутанные поводки собачек, получающих свободу под вашим руководством. Такова основная идея *TABT*: взять знакомую игру и подсластить ее сиропом.

У игры есть еще два режима: Test и Practice. В первом вы проходите пять игр, по одной на категорию, и в конце получаете оценку. Во втором режиме можно играть в любую «распечатанную» игру – и на время, и бесконечно (с увеличением сложности). Лучшие достижения можно даже опубликовать в Интернете.

В общем, *TABT* создана с любовью и заботой. У каждой игры собственная тема, графика пленяет мягкостью, а фоновая музыка услаждает слух (не отвлекая от игры). Прекрасный способ порадовать детей, заодно прививая им аналитические способности. Да и взрослому развлечься подчас совсем не грех (хотя игра, естественно, местами простовата). LXF





Режим Quest

Путешествуйте в поезде по карте, разгадывая загадки и открывая перегоны к новым играм.



Режим Practice

Играйте в те игры, что вы «распечатали» в режиме Quest (на время или просто так).



The Amazing Brain Train

Разработчик: Grubby Games
Сайт: http://grubbygames.com/tabt
Цена: $19,95

Функциональность	8/10
Графика	9/10
Увлекательность	7/10
Оправданность цены	10/10

» *Эффектная, милая, ласковая игра – развлекайтесь на здоровье!*



9/10

Picasa для Linux 2.7

Как получать изображения почти с любой цифровой камеры, редактировать их и организовать удобное хранение? **Дейл Стриклендт-Кларк** лишней суеты не любит…

Вкратце...

» Менеджер библиотеки изображений с базовыми функциями редактирования. Импортирует напрямую с цифровых аппаратов, читает большинство обычных RAW-форматов. Упрощает выгрузку изображений в Интернет. См. также: *DigiKam* и *F-Spot*.

Версия *Picasa* для Linux долго ходила в бедных родственниках у *Picasa* для Windows, но теперь разрыв сокращается. Релиз 2.7 получил много улучшений, среди которых поддержка широкого диапазона RAW-форматов (см. врезку внизу). Linux-версии все еще не достает некоторых функций – например, воспроизведения видео, автоматической записи DVD и полноценного показа слайд-шоу – но для большинства пользователей это несущественно. Главное, что *Picasa* – превосходный менеджер фотографий.

Свою работу программа делает не только безошибочно, но и изящно. В собственной базе данных *Picasa* изображения представлены миниатюрами, прокрутка которых немного инерционна. Переходы из одного состояния в другое происходят плавно: ничто не делается рывками, но ощущения медлительности нет. Фотографии остаются в тех папках, где их обнаружила *Picasa*, но при этом свободно группируются по альбомам, и любую фотографию можно включить в сколько угодно альбомов.

Изображения можно помечать подписями и ключевыми словами (сама программа не дает приоритета ни одному из двух вариантов). Подписи отображаются при просмотре миниатюр, а ключевые слова предназначены для меню. Для изображений в формате JPEG подписи и ключевые слова записываются непосредственно в файлы как IPTC-тэги, которые понимает практически все графическое ПО. Файлы других форматов *Picasa* обновлять не умеет, поэтому сведения о них записываются в **Picasa.ini** в той директории, где хранятся изображения.

› При просмотре миниатюр папки сортируются в алфавитном порядке, по дате создания или в виде дерева.

Командир картинок

Двойной щелчок на фотографии открывает встроенный редактор – в нем можно отрегулировать баланс белого и контраст, кадрировать картинку, добавить некоторые эффекты. Инструментарий ограничен самым необходимым, причем использование каждой функции упрощено до предела. Если разобраться в настройках редактора не представляется возможным, просто нажмите кнопку 'I'm Feeling Lucky' [Мне повезет!], и *Picasa* самостоятельно предпримет шаги, которые «посчитает разумными». Если сделанное изменение оказалось неудачным, его можно отменить в любое время, даже если закрыть программу и запустить снова несколько дней спустя. Ведь на самом деле *Picasa* не вмешивается в основной файл изображения, а только записывает сделанные изменения в **Picasa.ini** и применяет их всякий раз, когда файл открывается для просмотра. Получить измененную копию изображения можно с помощью кнопки Export [Экспорт].

Программа идеальна для предварительного просмотра RAW-файлов перед пост-процессингом с помощью программ вроде *UFRaw*. Иногда стандартная настройка уровней настолько совершенна, что вручную уже ничего не улучшить. А вот записывать *Picasa* может лишь в JPEG; это неприемлемо для профессиональной обработки фотографий, зато идеально для публикации в web. Сделать это в новой версии стало еще проще. Если у вас есть учетная запись Google Picasaweb (на http://picasaweb.google.com), то выгрузить свои альбомы в Интернет можно парой щелчков мыши, а потом – показывайте их хоть всему миру, хоть избранным приятелям. Профессиональные фотографы пользуются этой службой, чтобы клиент мог выбрать изображение для пост-процессинга или печати. LXF

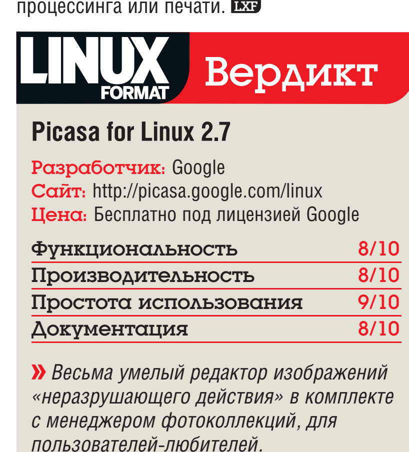
Что такое RAW-форматы?

Слово RAW (англ. «сырой», необработанный) относится к группе файлов, которые создаются большинством цифровых фотоаппаратов профессионального уровня. Диапазон форматов довольно широк, но одно их объединяет – все это данные, которые снимаются прямо с сенсора устройства, с минимальной аппаратной обработкой или вовсе без нее. В отличие от JPEG, «сырой» файл более гибок: он не подвергался сжатию, наводке на резкость, сглаживанию, подгонке баланса белого или корректировке насыщенности цветов.

Обычно цвет в пикселя в RAW-файле кодируется 12- или 14-битным числом (в JPEG для этого отводится всего восемь бит). В результате для каждого цвета становятся доступны не 256, а 4096 оттенков, что здорово влияет на отображение полутонов. Чтобы получить из RAW-файла различимое изображение, его нужно обработать (провести пост-процессинг). Обработка жестко контролируется и позволяет подогнать баланс белого, уровни цветовой насыщенности, яркости и контраста.

Многих фотографов привлекает степень свободы, предоставляемая подобными форматами файлов, но есть у них и недостатки. RAW-форматы отчасти проприетарны (некоторые даже кусочно зашифрованы, в попытке отогнать сторонних разработчиков), а по размеру они до шести раз превышают аналогичные JPEG-файлы. Для конвертации таких файлов необходимо специальное ПО, к тому же преобразования и редактирование обычно проводятся в двух разных программах.

Adobe пытается устранить эти проблемы с помощью собственного формата DNG [digital negative – цифровой негатив], но о сколь-нибудь серьезном успехе говорить пока рано.



LINUX FORMAT Вердикт

Picasa for Linux 2.7

Разработчик: Google
Сайт: http://picasa.google.com/linux
Цена: Бесплатно под лицензией Google

Функциональность	8/10
Производительность	8/10
Простота использования	9/10
Документация	8/10

» *Весьма умелый редактор изображений «неразрушающего действия» в комплекте с менеджером фотоколлекций, для пользователей-любителей.*

Рейтинг 8/10

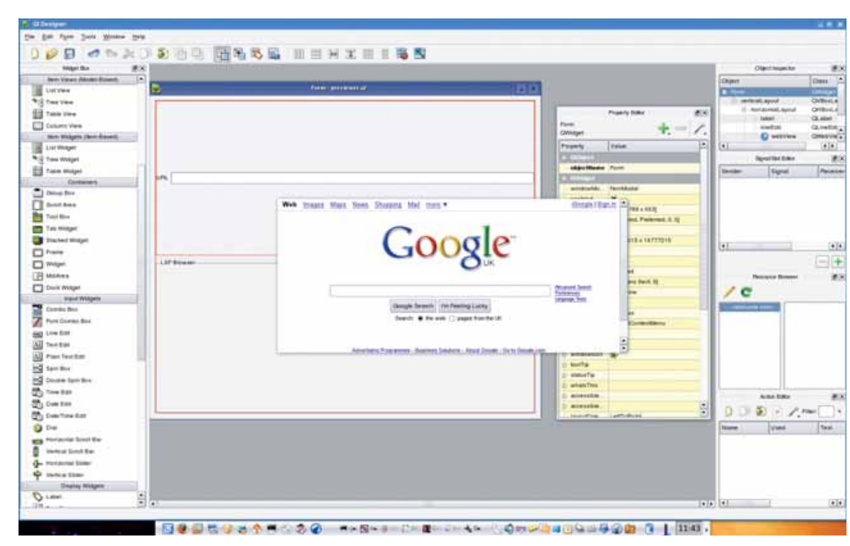
Qt 4.4 Open Source Edition

Коммерческие лицензии стоят тысячи в любой валюте, а **Грэм Моррисон** изучает GPL-версию, которая делает то же самое, но бесплатно…

Вкратце...

» Комплект программиста на C++ для разных ОС. Одновременно служит API для KDE. Среди альтернатив – Mono и GTK+.

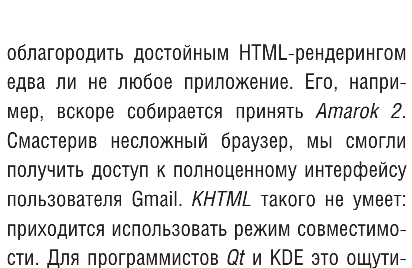
Вообще-то *Qt* – инструментарий программиста, используемый для создания кроссплатформенных приложений. Но это вовсе не значит, что выход новой версии интересен только узким специалистам. Многолетний союз намертво спаял *Qt* с KDE, и одно теперь немыслимо без другого. Новое в *Qt* неизбежно приводит к обновлению KDE, и эта версия не стала исключением. Надвигающийся и долгожданный KDE 4.1 базируется на *Qt 4.4*, и это здорово. *Qt 4.4* обладает многими замечательными качествами, и лучшее из них – включение *WebKit*, движка HTML-рендеринга, применяемого и *Safari* от Apple, и Series 60 от Nokia, и проектом Android от Google. Путь *WebKit* на рабочий стол Linux был долгим и нелегким.



» **Дизайнер интерфейсов *Qt* немало выиграл от обновления; перетаскивание живого виджета WebKit впечатляет.**

WebKit

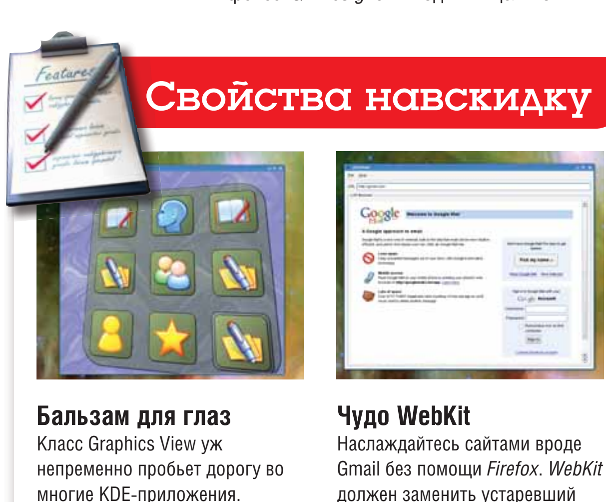
Появившись как «яблочная» ветка *KHTML*, *WebKit* вскоре стал основой собственного браузера и движка HTML-рендеринга Apple. В процессе разработки фирма внесла столько изменений, что они уже не вписывались в исходный проект *KHTML*, и он остался на обочине. К счастью, «смутное время» позади, и теперь *WebKit* входит в *Qt*. Во всех версиях *Qt* используется одна и та же реализация *WebKit*, что обеспечивает безупречную совместимость: все новое, что Apple делает для *Safari*, будет, благодаря *WebKit*, работать в *Qt*, а это вовлекает в оборот миллионы тестеров. Не факт, что *WebKit* станет движком рендеринга для *Konqueror* в KDE 4.1, но в следующих версиях KDE его влияние, несомненно, проявится – ведь он так удобен. Теперь можно просто перетащить виджет *WebKit* в интерфейсе *Qt Designer* и одним щелчком мыши облагородить достойным HTML-рендерингом едва ли не любое приложение. Его, например, вскоре собирается принять *Amarok 2*. Смастерив несложный браузер, мы смогли получить доступ к полноценному интерфейсу пользователя Gmail. *KHTML* такого не умеет: приходится использовать режим совместимости. Для программистов *Qt* и KDE это ощутимое подспорье.



Phonon

Еще одно приятное дополнение, связанное с KDE – это включение Phonon. Для тех, кто не в силах уследить за именами всех новых технологий KDE, Phonon – это мультимедиа-среда KDE. Превосходное дополнение с точки зрения программиста. Теперь для работы с музыкой и видео появился такой же удобный инструмент, который уже существовал для графики – он избавляет от неприятной рутины. Его можно встроить в свое новое приложение *Qt*, написав какие-то пять строчек кода. Причем эти пять строчек будут действовать одинаково в Windows, OS X и Linux – *DirectShow*, *QuickTime* или *GStreamer* выбираются автоматически, в зависимости от системы. Теперь взаимообмен мультимедиа-приложениями между Linux и другими платформами станет интенсивнее, а это одна из принципиальных задач KDE 4. Разумеется, в этой версии немало и других новинок. Например, в *QGraphicsView* теперь можно встраивать виджеты. Расширена поддержка XML, улучшена параллельная разработка, обновлены системы печати и справки. Все это внушает почтение к *Qt 4.4*, особенно если учесть, что библиотека выпускается по лицензии GPL. *Qt* всегда был кроссплатформенным API. С ним можно написать программу, способную работать повсюду, включая встраиваемые Linux-устройства и даже Windows Mobile. Иногда *Qt* запутан, громоздок, сложен для понимания. И все же на сегодня это лучший инструмент для кроссплатформенного программирования в мире свободного ПО. LXF

Свойства навскидку



Бальзам для глаз
Класс Graphics View уж непременно пробьет дорогу во многие KDE-приложения.

Чудо WebKit
Наслаждайтесь сайтами вроде Gmail без помощи *Firefox*. *WebKit* должен заменить устаревший *KHTML*.



LINUX FORMAT Вердикт

Qt 4.4 Open Source Edition

Разработчик: Trolltech
Сайт: www.trolltech.com
Цена: Бесплатно GPL

Функциональность	10/10
Производительность	7/10
Простота использования	7/10
Документация	9/10

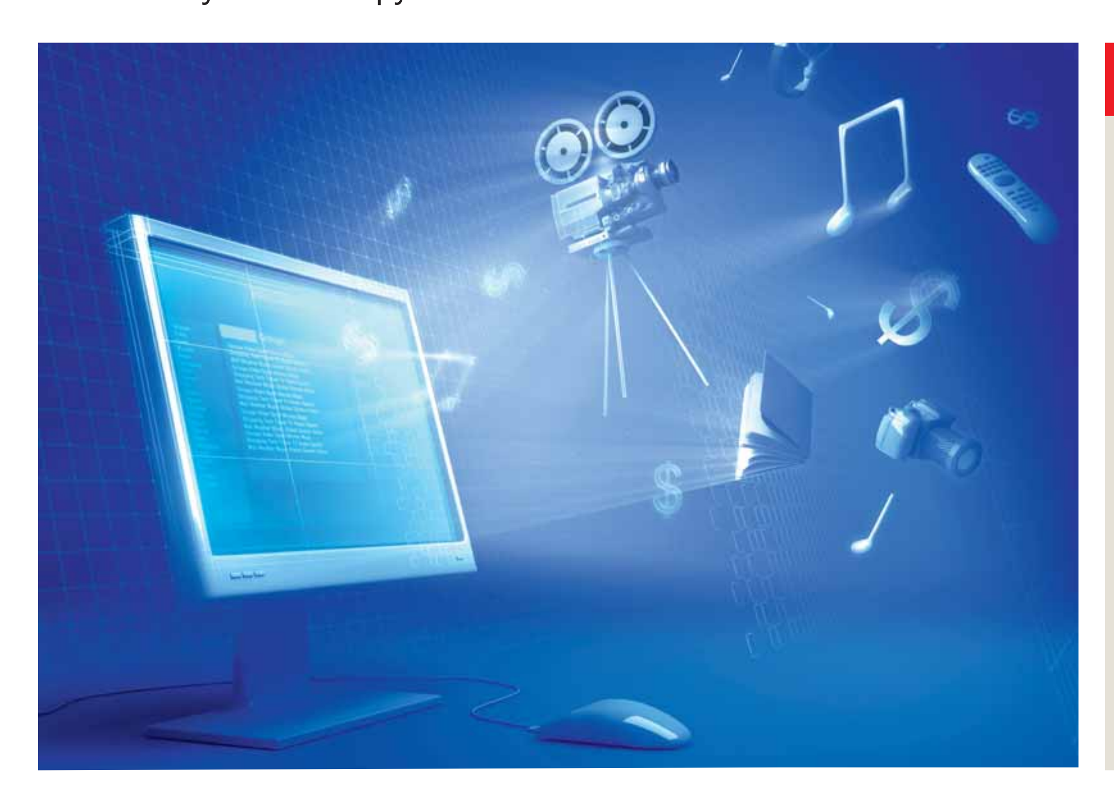
» *Хорош для пользователей KDE, но еще лучше для тех, кто пишет кроссплатформенные приложения с открытым исходным кодом.*

Рейтинг 8/10

» Каждый месяц мы сравниваем тысячи программ — а вы можете отдыхать!

Менеджеры закачек

Грэм Моррисон, ненасытная пиявка на нашем локальном сервере, отобрал для вас семь лучших инструментов. Готовьтесь стать качком…



Про наш тест...

В идеале, менеджер закачек должен повышать эффективность загрузки, оставаясь простым и удобным. Если он будет сложным, вряд ли кто-то захочет рыться в куче настроек только для того, чтобы добавить файл в очередь загрузки. Все это мы учли при оценке.

Самые «умелые» менеджеры обычно сложны в изучении, что отпугивает многих потенциальных пользователей. Поэтому первенство будет отдано программе, сумевшей выдержать баланс между удобством и функциональностью. Ускорить загрузку хотят все, но долго думать по этому поводу некогда. Мы оценивали семь наших претендентов по нескольким критериям, среди которых простота использования, эффективность загрузки и удобство организации интерфейса (даже если это инструмент командной строки).

Наш выбор

Менеджер закачек выполняет две основных функции. Во-первых, он упорядочивает загрузку файлов, собирая их в единое, централизованное окно на рабочем столе. Во-вторых, он повышает скорость скачивания.

Упорядочение очень важно, если приходится загружать много файлов из разных источников. Имеются в виду те вещи, на которые мы «набредаем» во время интернет-серфинга: для BitTorrent и других P2P-протоколов существуют специализированные программы. И Gnome, и KDE открывают для каждого файла, копируемого с удаленного хоста, отдельное окно. У *Firefox* есть встроенный менеджер, но с его помощью можно лишь отменить загрузку файла или очистить всю очередь [в версии 3.0 ситуация изменилась, – *прим. ред.*]. Инструменты, которые мы рассмотрим, позволяют собирать все ссылки в едином окне, приостанавливать и возобновлять загрузку, переупорядочивать очередь, а также поддерживают параллельную и сегментированную закачку.

Демон скорости

Сегментация и параллельная загрузка ускоряют процесс скачивания, расшивая два узких места на линии «клиент-сервер». Многие серверы ограничивают трафик по подключениям, чтобы единственный пользователь не занял весь канал. Разделив файл на части и используя для загрузки каждой из них отдельное соединение, данное ограничение можно обойти. Быть может, это не совсем этично, так как ваше множественное подключение перекроет доступ другим людям [поэтому администратор на той стороне может решить проблему радикально, заблокировав ваш IP, – *прим. ред.*], но если сервер не перегружен запросами, скорость загрузки существенно повышается. В таком случае параллельное скачивание бывает очень полезно, особенно если речь идет о новом дистрибутиве. Секции файла можно брать с разных серверов, распределяя нагрузку между несколькими машинами. Это идеально для дистрибутивов Linux, которые обычно выкладываются на ряде зеркал.

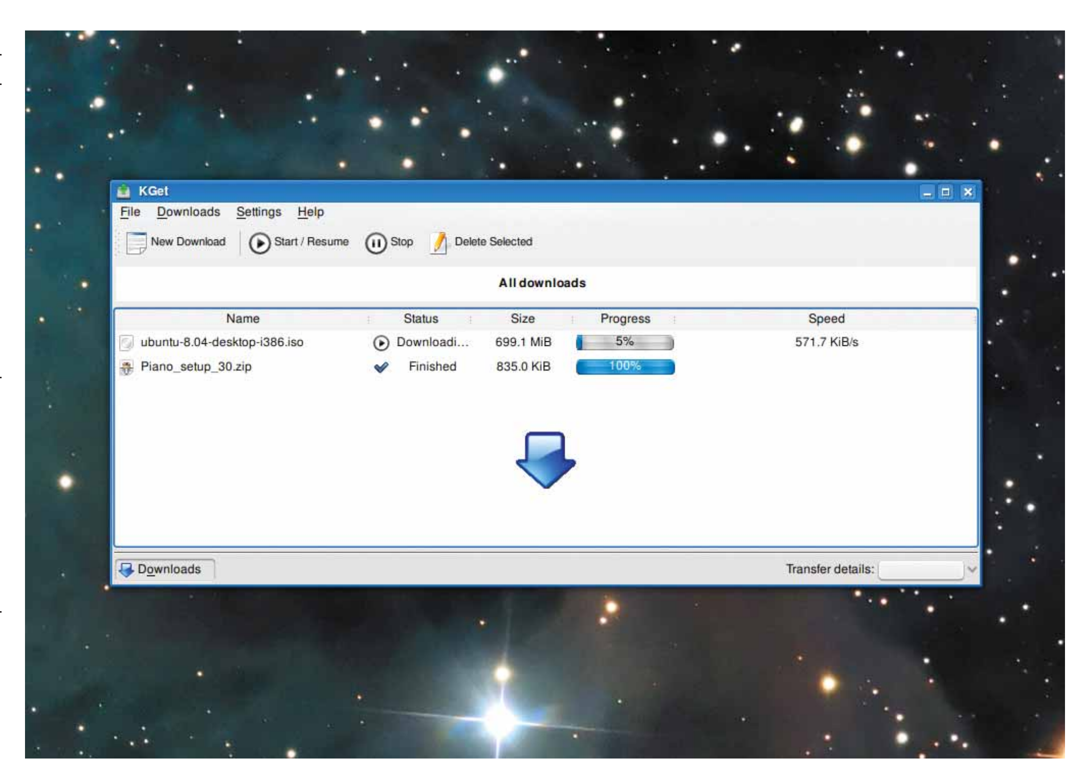
KGet

Пользователи KDE могут расслабиться: прекрасный менеджер закачек у них уже есть. Прямо в меню «Программы».

KGet неотделим от KDE. Его функциональность тесно переплелась с данным рабочим столом и с *Konqueror*. И если у вас нет ни того, ни другого, толку от *KGet* мало. С другой стороны, если вы KDE-пользователь, такая интеграция может перевесить самые замечательные функции остальных менеджеров закачек.

Поэтому удивляет, что по умолчанию *KGet* устанавливается редко. Почти всегда его нужно инсталлировать вручную, через менеджер пакетов. При первом запуске программы вам предлагается решить, согласны ли вы сделать *KGet* менеджером закачек *Konqueror*. Предсказуемый ответ «да» – и *KGet* прозрачно интегрируется в *Konq*.

Щелчок на ссылке в *Konqueror* отправляет запрос на закачку в *KGet*, тот добавляет файл в очередь и загружает его. Рабочий стол не загромождается: ведь все загружаемые файлы группируются в одном окне.



› Загружать можно только то, что вам нужно — KDE-пользователям повезло.

Как в аптеке

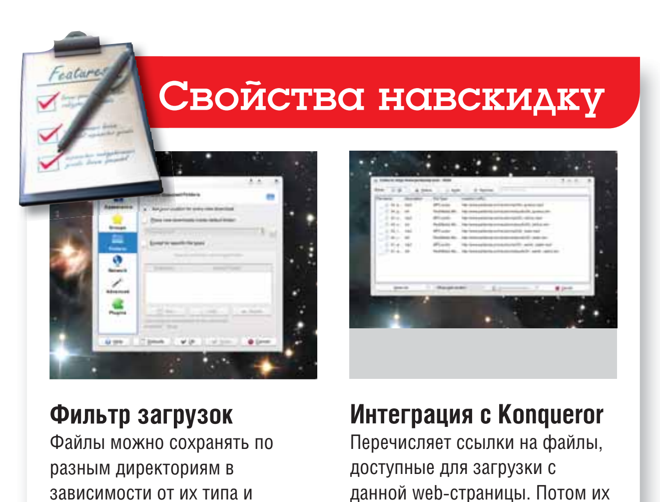
Именно в нем можно приостанавливать или возобновлять закачки (хоть скопом, хоть по одной). Загрузка автоматически продолжается после разрыва связи или выходе пользователя из системы (когда он вернется обратно). В простом окне прекрасно видно состояние каждой загрузки: скорость и потребляемый трафик.

Еще одно изменение после инсталляции *KGet* – появление в меню *Konqueror* новых пунктов, Show Drop Target (отображать корзину закачек) и List All Links (выводить все ссылки). Вообще-то первый пункт не только для меню: это общая команда, которая доступна также по щелчку на значке *KGet* в системном лотке. При ее активации на рабочем столе появится широкая синяя стрелка, на которую можно перетаскивать URL файлов для загрузки по протоколам HTTP или FTP прямо из окна браузера (включая *Firefox*). *KGet* добросовестно скачает выбранные файлы в фоновом режиме. По умолчанию программа задает вопрос, где сохранить загруженное, но в окне настройки нетрудно определить иное поведение. Приятное дополнение – возможность сортировать файлы в соответствии с их типом. Картинки автоматически отправляются в одну папку, музыка – в другую.

А самое интересное мы приберегли напоследок. Вторая дополнительная команда в меню *Konqueror* – это просто находка. Выберите List All Links – откроется окно, в котором будут перечислены все файлы, доступные к загрузке с текущей web-страницы. Это бесценно, если вам часто приходится скачивать кучу файлов с одного сайта. Например, это может быть главный сайт проекта, пестрящий дополнительными пакетами и зависимостями.

Окно List All Links позволяет сразу выбрать то, что необходимо загрузить, вместо того, чтобы щелкать на каждой ссылке в отдельности. Нажмите кнопку Audio, и увидите все музыкальные файлы. Наберите в поле поиска '*.mp3' – перед вами появятся все доступные MP3. Это уникальное свойство *KGet*, поэтому его установка насущно необходима KDE-пользователю, если он качает часто и помногу.

Свойства навскидку



Фильтр загрузок
Файлы можно сохранять по разным директориям в зависимости от их типа и названия.

Интеграция с Konqueror
Перечисляет ссылки на файлы, доступные для загрузки с данной web-страницы. Потом их можно добавить в очередь.

LINUX FORMAT Вердикт

KGet

Версия: 2.03 (KDE 4); 0.8.5 (KDE 3.5)
Сайт: http://kget.sourceforge.net
Лицензия: GPL

» Отличный выбор для KDE-пользователя. Не хватает только средств ускорения загрузки.

Рейтинг 8/10

Wget

Может ли первая ласточка остаться лидером?

Да, мы имеем в виду утилиту командной строки, которую часто приходится устанавливать вручную и чья история восходит ко временам, когда Всемирная паутина была крохотной паутинкой. *Wget* сегодня актуален так же, как в далеком 1996-м, когда наши прапрадеды прокладывали первые интернет-соединения. В ряде случаев *Wget* был единственным менеджером из нашего **Сравнения**, сумевшим восстановить нарушенную конфигурацию *X* и загрузить для нас видеодрайверы Nvidia.

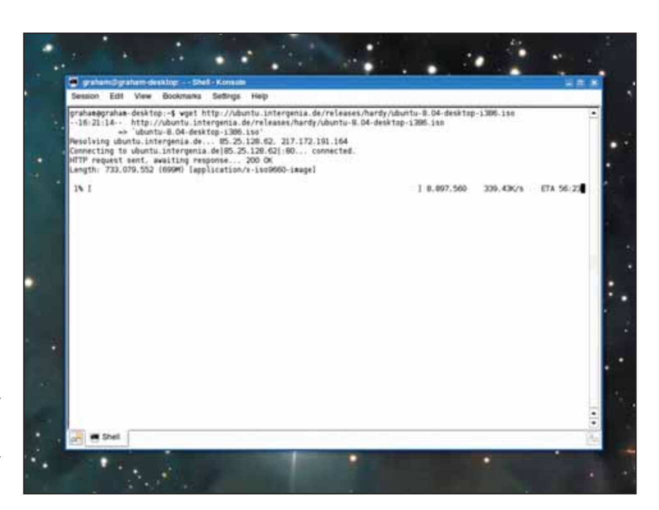
Пользоваться программой сверхпросто. Стоит набрать '*Wget*' и прибавить URL выбранного файла, как *Wget* тут же скопирует все в текущую директорию. Пользовательский интерфейс для инструмента командной строки совсем неплох. Вдоль терминала бежит полоска индикатора загрузки, которая перескакивает на новую строку по достижении определенных отметок в загружаемом файле.

Освоить *Wget* нетрудно, причем программа четко отображает текущие параметры закачки (скорость и расчетное время окончания). Можно даже приостановить загрузку и начать ее заново, возобновить прерванную закачку и перевести работу в фон, передав ключ --background во время выполнения.

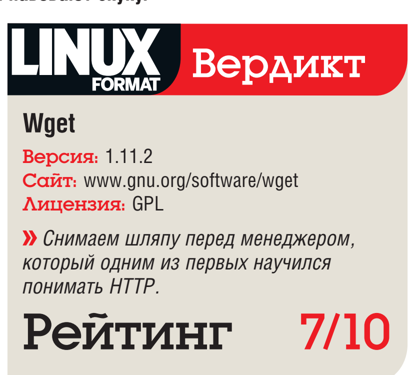
И это не предел способностей *Wget*. С его помощью можно зеркалировать и загружать сайты целиком. Ввод аргументов -r -l 0 приведет к скачиванию сайта в полном объеме, с анализом HTML и загрузкой всего необходимого для воссоздания ресурса на локальной машине. Программа идеальна для автоматического резервирования и использования в простых скриптах. Например, аргумент -N позволяет закачивать только те файлы, онлайн-копии которых созданы позже локальных.

Если подключение к серверу проходит по SSH, то пользоваться *Wget* гораздо проще, чем FTP, к тому же *Wget* присуща замечательная гибкость в отношении типов загружаемых файлов.

> «Только Wget сумел восстановить нарушенную конфигурацию X.»



› Проблема программ командной строки (даже самых крутых) в том, что их экранные снимки навевают скуку.



LINUX FORMAT Вердикт

Wget

Версия: 1.11.2
Сайт: www.gnu.org/software/wget
Лицензия: GPL

» *Снимаем шляпу перед менеджером, который одним из первых научился понимать HTTP.*

Рейтинг 7/10

GWget

Что бы такое значила буква 'G' в названии?

Как следует из имени, это версия *Wget* для графического окружения Gnome: она разработана в помощь обычным людям, нуждающимся в способностях *Wget*, но не склонным работать в бородатом терминале образца 70-х.

И программа прекрасно справляется со своей задачей. Щелкните по кнопке New, введите URL файла; *GWget* его тут же загрузит. А можно перетащить ссылку из *Firefox* в главное окно, и файл добавится в очередь. Когда-то было даже особое расширение для *Firefox*, помогавшее отсылать файлы в очередь *GWget*, но в последнее время оно как-то затерялось: ни во 2-й, ни в 3-й версии его уже нет. Но не все так плохо – интеграция с *Epiphany* осталась. Это встроенный браузер Gnome, он использует тот же движок, что и *Firefox* (пока что). Можно установить расширение *Epiphany*, интегрирующее *GWget* с браузером (подобно связке *Konqueror* и *KGet*). Поскольку функциональность *GWget* определяется *Wget*, поддерживаются только HTTP и FTP, но простая страничка настройки позволяет создавать рекурсивные закачки, настраивать прокси-серверы и ограничивать скорость загрузки. Есть и графическое отображение очереди закачек – еще одно приятное дополнение. В окне выводятся скорость и состояние задачи, а также расчетное время ее завершения. Здесь же можно отменить ставшую ненужной загрузку и сгенерировать MD5-сумму: эта функция имеется только у одного из менеджеров в нашем **Сравнении**.

Если вам нравится гениальная простота *Wget*, но не по вкусу интерфейс командной строки, к *GWget* стоит присмотреться. К сожалению, средства ускорения загрузки у него отсутствуют напрочь.

> «Выводятся скорость, состояние и время завершения загрузки.»

› Одна из особенностей, «унаследованных» *GWget* от *Wget* – дизайн графического интерфейса в духе Марка Ротко [Mark Rothko].



LINUX FORMAT Вердикт

GWget

Версия: 0.99
Сайт: www.gnome.org/projects/gwget
Лицензия: GPL

» *Это Wget в блестящей обертке Gnome-интерфейса, но есть и настольные альтернативы получше.*

Рейтинг 6/10

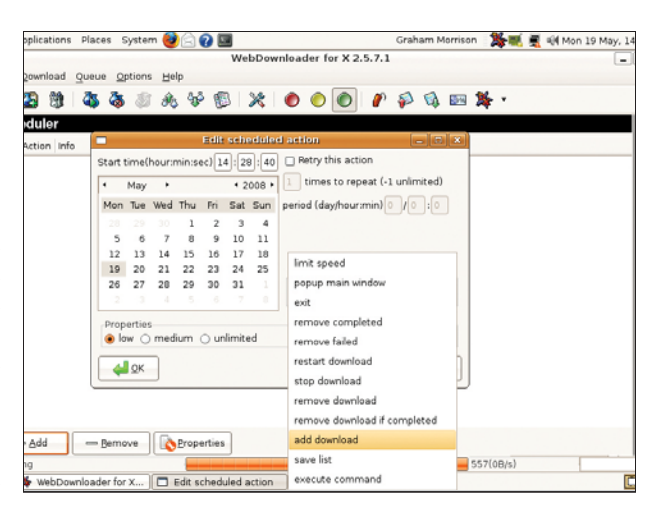
Downloader for X

Всякий ли новичок знает, что такое X?

Downloader изрядно смахивает на менеджеры закачек для Windows. Значительная часть окна отводится отображению статуса загрузки. Как и в *KGet*, здесь выводятся все файлы из очереди закачек, вместе со статистикой (скорость, примерное время завершения и пр.). *Downloader* также поддерживает FTP- и HTTP-подключения, а файлы можно добавлять кнопкой New Download или копированием URL через буфер обмена. Если активировать иконку «корзины закачек», можно перетаскивать в нее файлы из браузеров и файловых менеджеров. Масками в названиях файлов можно задать фильтр – очень полезное дополнение, если с FTP загружается целая серия однотипных файлов.

Downloader анализирует HTML и загружает файлы, на которые ведут ссылки с web-страниц. Возможна рекурсивная загрузка. Параметры закачки настраиваются в окне Edit Download. Здесь можно установить всевозможные опции: от FTP-пароля до количества сегментов, скачиваемых с сервера одновременно. Интересна панель автоматизации. В ней устанавливается расписание загрузки по определенным времени и дате. Есть и другие параметры настройки: например, ограничение скорости загрузки, выполнение определенных команд и очистка очереди закачек. Если что-то пойдет не так, к вашим услугам журнал загрузок, разноцветные надписи которого помогут разобраться с неправильными масками или проследить битые ссылки. Симпатичная программа, но *Downloader* тоже страдает отсутствием ускорителей загрузки.

> «Для загрузки серии файлов с FTP можно пользоваться масками.»



› Кто пользовался менеджером закачек для Windows, в *Downloader for X* почувствует себя как дома.



LINUX FORMAT Вердикт

Downloader for X

Версия: 2.5.7.1
Сайт: www.krasu.ru/soft/chuchelo
Лицензия: GPL

» *Достаточно удобен для менеджера средних способностей с работой по расписанию. Однако не оригинален.*

Рейтинг 7/10

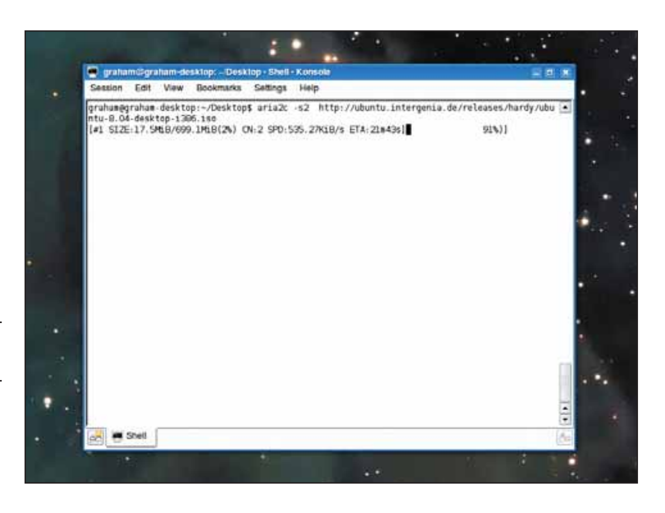
Aria2

Японо-итальянские напевы...

Для маленьких, но мощных и эффективных программ Linux – превосходная платформа. Именно такой мы нашли *Aria2*. На первый взгляд она мало чем отличается от *Wget*: тот же терминал, после команды набирается URL, так же загружается файл. Но *Aria2* представляет более развитое поколение менеджеров закачек. Например, добавив аргумент -s2 к начальной команде, можно разбить соединение на два. Если сервер ограничивает трафик на подключение, это удвоит скорость. Есть еще одна уловка: можно использовать два URL, указывающих на один и тот же файл, а после загрузки *Aria2* соединит данные из обоих источников. Можно даже разделить канал связи между протоколами FTP и HTTP для большей гибкости. Из разных сегментов программа соберет готовый файл.

За последние несколько лет HTTP и FTP начали сдавать позиции BitTorrent, и это не укрылось от автора *Aria2* Тацухиро Цуджикава [Tatsuhiro Tsujikawa]. Достаточно набрать '*Aria2*c file.torrent', и файлы из указанного torrent'а будут автоматически загружены. Можно ограничить трафик загрузки и выгрузки, указать torrent-файл на удаленном сервере без предварительного скачивания. Можно даже изменить соотношение загрузки/раздачи или время, затрачиваемое на «сидинг» – то есть отдать столько же, сколько получено, не подвергая канал излишней нагрузке. Да, инструменты командной строки, как говорится, «на любителя», поэтому есть попытки разработать графическую оболочку для *Aria2*, но программа и без того достаточно проста. И работает гораздо быстрее графических менеджеров.

> «Aria2 — это новое, продвинутое поколение менеджеров закачек».



› Смотреть особо не на что, но при этом *Aria2* грузит ISO Ubuntu в два сегмента на двойной скорости.



LINUX FORMAT Вердикт

Aria2

Версия: 0.13.1+2
Сайт: http://aria2.sourceforge.net
Лицензия: GPL

» *Хотя это инструмент командной строки, менеджер закачек Aria2 очень прост в использовании и суперсбыстр.*

Рейтинг 9/10

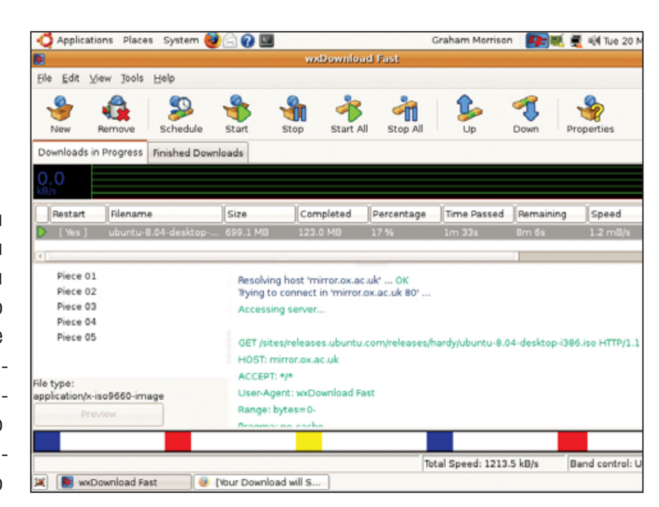
WXDownload Fast

Милости просим в страну дурацких названий.

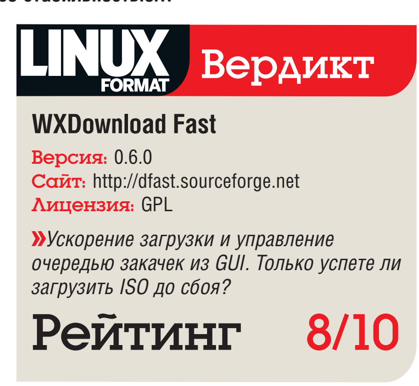
Любой менеджер, отважившийся выставить график производительности напоказ, стоит внимания. *WXDownload Fast* как раз из таких. График проходит прямо по середине главного окна. Во время загрузки отображается суммарная текущая скорость всех закачек. На чуть-чуть это показуха, но позволяет судить о ширине канала.

И скорость – явно первейшая забота программы. Закачки добавляются выбором URL в браузере или почтовом клиенте и последующим нажатием кнопки New в окне '*WXDFast*' (так программу тоже можно называть). Можно указать URL нескольких зеркал одного файла, или разделить его на отдельные сегменты и загружать их параллельно – потенциал скорости закачки увеличивается многократно. График позволяет оценить разницу между сегментированной и обыкновенной загрузками. Правда, это нетрудно сделать и в остальных менеджерах, просто сравнив скорости. Если вы хотите приберечь часть трафика для просмотра YouTube, можно ограничить интенсивность загрузки или даже составить расписание закачек по дате и времени. Все эти функции собраны в простом и приветливом интерфейсе пользователя, и это единственное приложение, в котором параметры сегментирования закачек можно задать в GUI. Но это и единственное приложение во всем **Сравнении**, у которого наблюдались проблемы со стабильностью. Добавление некоторых URL приводило к краху программы, а поскольку адреса записываются в файл конфигурации, после этого невозможно было даже запустить *WXDFast*: при попытке возобновить прерванное соединение сбой повторялся. Единственный выход в такой ситуации – удалить файл с опасными URL и запустить программу снова.

> «Первейшая забота программы — явно скорость».



› В отношении скорости и удобства *WXDownload Fast* бесподобен. Если бы не проблемы со стабильностью...



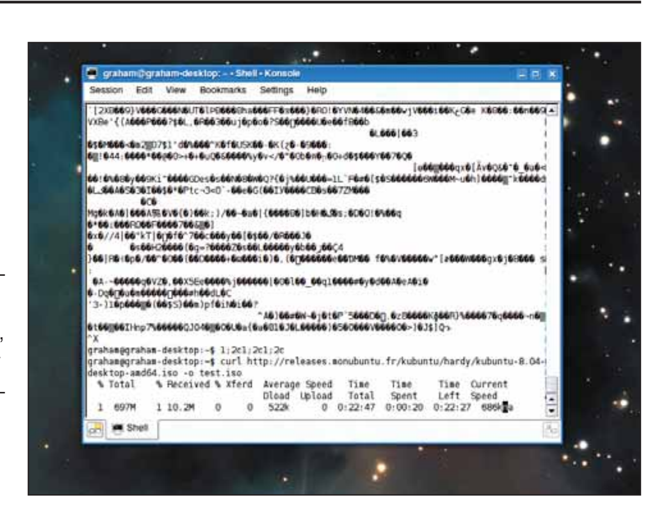
Curl

Загрузка в стиле «пленных не брать».

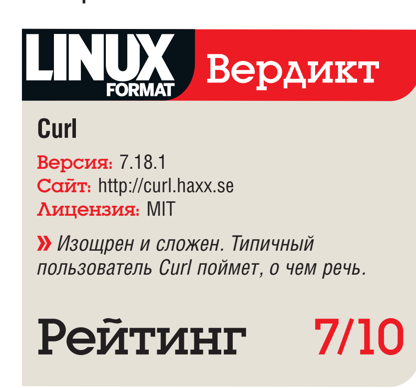
Если вы думаете, что *Aria2* и *Wget* сложны – так это вы еще *Curl* не пробовали. Здесь тот же интерфейс командной строки, к которому мы уже начали привыкать: набираем curl в сопровождении URL файла, и загрузка начинается автоматически. Но, в отличие от *Aria2* и *Wget*, *Curl* выводит скачиваемое прямо на консоль, вмиг покрывая экран кракозябрами и загогулинами. Выход есть: перенаправьте вывод в отдельный файл, как это делается с командой cat. Можно набрать аргумент -o и указать выходной файл; в качестве бонуса отображается индикатор прогресса.

За внешней неприветливостью кроется подлинная «крутизна»: *Curl* – это инструмент программиста. По духу программа ближе к Unix, чем все остальные менеджеры. Сопутствующие библиотеки используются многими разработчиками для встраивания загрузочных функций в их приложения: модули *Curl* существуют для PHP, файловых систем FTP, Python и десятков других инструментов. Вот эта бездна параметров и протоколов нас и отпугнула. HTTP, HTTPS, FTP, FTPS, SCP SFTP, TFTP, DICT, Telnet и LDAP – поддерживается все. Наш коллективный разум так и не придумал, зачем при загрузке файлов DICT, Telnet или LDAP, но если вы не согласны – что ж, каждому свое. Поддержки сегментации и параллельной загрузки нет, рекурсивное копирование серверов не предусмотрено, анализ HTML не проводится. Правда, скрипт под названием *mcurl* позволяет разбить файл на несколько сегментов, и *Curl* также можно использовать для выгрузки файлов. Но если брать в целом, то этот менеджер закачек больше годится для автоматизации загрузки с помощью пользовательских скриптов.

> «Нас отпугнула бездна параметров и протоколов.»



› Если экран монитора вдруг покрылся каббалистическими знаками – вспомните азбуку консольной грамотности.



Менеджеры закачек

Вердикт

Aria2 9/10

Придти к заключению в этом **Сравнении** было очень непросто. Наш победитель – инструмент командной строки с парой графических оболочек, ни одну из которых рекомендовать пока нельзя. Не таким должен быть лауреат. Но ведь менеджер закачек оценивается по эффективности загрузки и удобству использования. Пусть *Aria2* консольная – она вполне отвечает обоим критериям. Есть и еще одно обстоятельство, которое помогло ей взобраться на первое место. Почти все менеджеры закачек, рассмотренные в нашем Сравнении, могут интегрироваться с фантастическим расширением *Firefox* под названием *FlashGot*.

Это что-то вроде мета-менеджера закачек, менеджера менеджеров, и ему в подчинение можно отдать любую утилиту, установленную в системе – в том числе победителя, *Aria2*. А значит, пользователю *Firefox* нет нужды осваивать консоль. Назначаете *Aria2* менеджером закачек для *FlashGot* – и готово.

Второй эшелон

Другие участники конкурса тоже заслуживают упоминания: например, к победе был очень близок *KGet*. Его интеграцию с рабочим столом и функцию сортировки закачек превзойти трудно, да и в обращении он проще других.

Проблема в том, что преимущества *KGet* доступны лишь KDE-пользователям, да и тем процесс загрузки не ускорить. Уважаемые разработчики программы! Если вы читаете наш журнал и при этом еще не забросили свое детище, готовя долгожданный релиз 4.1 – пожалуйста, учтите, что сегментация и параллельная загрузка ну очень пригодились бы!

Другое приложение, *Curl*, замечательно тем, что используется многими сторонними программами. Если вы решили создать собственный скриптовый загрузчик или включить в свое приложение модуль для выгрузки, этот инструмент – для вас. Только не воображайте, что освоите его раньше чем через час внимательного чтения руководства.

И, наконец, если команде разработчиков *WXDownload Fast* удастся упрочить стабильность приложения, это будет самый дружелюбный сверхскоростной загрузчик. А пока – выбираем *Aria2*: она быстра, и она работает. LXF

Обратная связь

Говорят, лучше один раз увидеть, чем сто раз услышать. Попробуйте наши рекомендации, загружая новый дистрибутив, и поделитесь с нами мнением о том, что считаете лучшим именно вы. Вариантов очень много, и все они отвечают разным вкусам. Мы будем рады узнать, как вам удалось встроить любимый менеджер закачек в свой рабочий стол. Черкните строчку на letters@linuxformat.ru или выразите свою точку зрения на www.linuxforum.ru.

Скачайте FlashGot



Файлы для загрузки мы, в основном, находим во время просмотра web-страниц. Поэтому некоторые менеджеры закачек интегрируются со своими браузерами – *KGet* с *Konqueror*, *GWget* с *Epiphany*... Загрузка файлов здесь происходит без обращения к другим программам. В случае с Firefox все немного не так.

Расширения, которыми он так богат, исповедуют обратный подход: они добавляют функциональность основному браузеру, а не сторонней программе. Если вы ищете менеджер закачек, на сайте Mozilla можно найти десятки вариантов. Но немногие из этих программ обладают разнообразием функций и гибкостью, присущими участникам нашего Сравнения. Именно поэтому так популярен *FlashGot*. Его загружают 250 000 раз в неделю, и это превосходный выбор для Linux-пользователей, которые часто скачивают файлы. Зачем изобретать велосипед, если *FlashGot* подстроится почти к любому менеджеру закачек, включая версии для Windows (при наличии *WINE*). Итого: если у вас *Firefox*, то *Aria2* и *FlashGot* – идеальная комбинация.

Таблица свойств

	Докачка	Пауза	Потоки	Сегментация	Рекурсивная загрузка	HTTP	HTTPS	FTP	Рабочий стол	Фильтр	Metalinks	Torrents	Планирование	MD5
Aria2	✓	☒	✓	✓	✓	✓	✓	✓	☒	☒	✓	✓	☒	☒
Curl	☒	☒	✓	☒	☒	✓	☒	✓	☒	☒	☒	☒	☒	✓
Downloader for X	✓	✓	☒	☒	✓	✓	✓	✓	✓	✓	✓	☒	✓	☒
GWget	✓	✓	☒	☒	✓	✓	✓	✓	✓	✓	☒	☒	☒	☒
KGet	✓	✓	☒	☒	✓	✓	✓	✓	✓	✓	☒	☒	☒	☒
Wget	✓	✓	☒	☒	✓	✓	✓	✓	☒	☒	☒	☒	☒	☒
WXDownload Fast	✓	✓	✓	✓	✓	☒	✓	✓	✓	✓	☒	✓	✓	✓

Distrowatch



Ежемесячная сводка новостей дистрибутивов Linux.



СЬЮЗЕН ЛИНТОН
Сьюзен Линтон является владелицей и оператором tuxmachines.org.

Slackware – 15!

Slackware Мы желаем дистрибутиву-дедушке еще много раз отмечать день рождения: свой 15-й он встречает в версии 12.1.

Старейший коммерческий дистрибутив Linux отметил свой 15-й день рождения. 16 июля 1993 г. Патрик Фолькердинг [Patrick Volkerding] анонсировал первую версию Slackware – ответвление от исходного Softlanding Linux System, популярнейшего из всех ранних дистрибутивов Linux. С того исторического дня прошло пятнадцать лет – и сейчас в версии 12.1 Slackware столь же силен, как и всегда.

Slackware 12.1 поднимает систему на новый уровень. Особенно впечатляет степень модернизации функций: создается ощущение, что релиз современный и идет в ногу с прочими. Установщик – все тот же мастер *Ncurses*, почти ровесник самого дистрибутива. Единственное новшество – добавление настроек безопасности NTFS, позволяющих устанавливать права чтения/записи.



› Slackware 12.1 теперь поддерживает исключительно ядра серии 2.6.

Во внешнем виде также не заметно радикальных изменений, но это не недосмотр, а философия Slackware; она-то и делает дистрибутив таким удобным в использовании. Фолькердинг предоставляет то, что можно описать как стандарт Linux. Он упаковывает приложения для менеджера пакетов Slackware, но очень слабо меняет базовый код. Этим укрепляются стабильность и надежность; но и ожидания пользователя не обмануты. А потому, вероятно, верно утверждение, что Slackware – дистрибутив «без сюрпризов».

> «Если вы никогда или давно не работали в Slackware, пора начать.»

Другой аспект философии Slackware – быть проще. Лично мне это всегда импонировало. Один из примеров – конфигурация системы. В прошлом, для настройки своих устройств и интернет-соединения от пользователя требовалось отредактировать несколько файлов в определенном каталоге. Сегодня многое определяется и настраивается автоматически, как и в любом другом дистрибутиве, но для некоторых вещей Slackware все еще опрашивает эти файлы. После моей установки, добавление учетной записи пользователя, редактирование файла **rc.modules** для поддержки изменения частоты процессора и правка **xorg.conf** для указания драйвера и настройки желаемого разрешения заняли менее пяти минут.

Slackware 12.1 использует *X.org 7.3*, Linux 2.6.24-5 и *GCC 4.2.3*. Кроме стандартных приложений KDE, он также содержит *GIMP*, *Amarok*, *Audacious*, *Gxine*, *KWord*, *Firefox 2.0.0.14*, *Thunderbird*, *SeaMonkey*, *Pan*, *Pidgin* и *wpa_gui*. Все инструменты разработки и библиотеки обновлены: например, взяты версии Perl 5.8.8, Python 2.5.2 и *Glibc 2.7*.

Slackware поддерживает несколько мультимедийных форматов, таких как AVI, MPEG, MP3, MP4 и OGG. Сменные носители монтируются автоматически при помощи HAL и *udev*, а *KLaptop* выполняет полезную работу, предоставляя пользователю функции энергосбережения и мониторинга батарей.

В целом, этот релиз вводит Slackware в настоящее со всеми удобствами, имеющимися в других дистрибутивах Linux, сохраняя верность испытанной формуле. Если вы поклонник Slackware, то останетесь довольны. Если вы никогда или уже давно не работали в Slackware, вам пора это сделать. Это современный дистрибутив с древним сердцем. Так что поднимем бокалы и скажем «С днем рождения, Slackware!»
www.slackware.com

Sugar – не сахар

Последние несколько недель ходили слухи, что Windows XP, возможно, появится на ноутбуках XO-1 проекта «Ноутбук каждому ребенку» (OLPC). 15 мая это стало фактом. Согласно совместному пресс-релизу Microsoft и OLPC, XO-1 в итоге будет с двойной загрузкой Windows и Linux, предоставляя пользователю самому выбирать, что использовать. Многие увидели в этом еще одну попытку Microsoft застолбить грядущие позиции в развивающихся странах. Несомненно, так и есть, и образование самых нуждающихся детей в мире может пострадать от утраты преимуществ, предоставляемых Open Source и mesh-сетями.

Выигрывают все

В идеальном мире многие из нас предпочли бы, чтоб философские аргументы в пользу ОС на базе Sugar и против Microsoft победили, но реальность диктуется деньгами. Ноутбуки XO-1 с Sugar просто не продавались так хорошо, как предсказывалось, и раз многие страны просили Windows, что было делать некоммерческой организации? Основатель OLPC и ее глава Николас Негропонте [Nicholas Negroponte] сообщил *The New York Times*, что многие правительства, покупающие или задумывающие купить эти машины, просят Windows.

Если для того, чтобы эти ноутбуки попали к детям в развивающихся странах, непременно нужен Windows – пусть будет так. В конце концов, им лучше получить ноутбук с ОС от Microsoft, чем не получить его вовсе. Это один из тех случаев, когда политику и убеждения надо отложить в сторону. Речь не о том, что Microsoft выиграла, а Linux проиграл: единственно важный здесь фактор – то, что ноутбуки вообще будут распространяться.
susan.linton@futurenet.com

Mini Mepis

Antix Легковесный, но полноценный CD-дистрибутив.

Antix – это меньшая, облегченная версия SimplyMepis, разработанная для запуска на старых машинах. С ним справится и ПК с процессором PII-266 МГц и 64 МБ ОЗУ, но рекомендуется 128 МБ. Разработанный с прицелом на слабые компьютеры, он тем не менее достаточно хорошо оснащен для повседневных нужд.

Antix M7.2 – самый последний релиз, основанный на Mepis 7.0. Менеджер окон в нем – *Fluxbox*, но доступен и IceWM. Разработчики выбрали более легкие приложения для обеспечения производительности не в ущерб функциональности. Среди них – *Iceweasel 2.0.0.14*, *Dillo*, *Claws Mail*, *Pidgin*, *Transmission*, *AbiWord*, *Gnumeric*, *Osmo*, *MTPaint*, *GQview*, *Brasero*, *MPlayer*, *XMMS* и *Conky*. Antix использует ядро 2.6.22-1 Mepis, *X.org 7.3* и *GCC 4.3.1*.

Будучи основанным на Mepis, Antix использует систему управления пакетами на базе APT и включает графическую оболочку *Synaptic*. Он также же унаследовал от Mepis множество инструментов администрирования: настройку сети, принтеров и пользователей, а также проприетарные драйвера видеокарт для чипов ATI (AMD) и NVIDIA. Mepis и Antix даже содержат драйвера для некоторых закрытых устройств, не поддерживаемых ядром Linux: например, карт беспроводной связи Broadcom.



› Рабочий стол Antix использует компактный, быстрый и легкий оконный менеджер *Fluxbox*.

Самый старый компьютер, на котором мы тестировали Antix, был PIII-667 с 384 МБ ОЗУ, и там он работал отлично. Многие другие современные дистрибутивы кажутся тяжелыми для таких старых машин, особенно с KDE или Gnome, но Antix делает их использование приятным. Он будет так же хорошо работать и на современном ноутбуке, предоставляя поддержку изменения частоты процессора, опции слежения за батареей и засыпания, включенные по умолчанию. И он хорошо выглядит!

http://antix.mepis.org

Мал, да удал

Puppy Linux 4.0 Могучие 88 МБ солидной компьютерной силы.

Puppy Linux всегда умел произвести впечатление. При крошечном объеме загрузки и достаточной легкости, чтобы запуститься на Pentium-166 MMX с 32 МБ ОЗУ (рекомендуется 128 МБ), он обладает всеми функциями нормального ежедневного рабочего стола. Последний выпуск 4.0 Dingo даже лучше, чем прежде.

С версией 4.0, вы не жертвуете красотой ради функциональности: разработчики адаптировали GTK2 и добавили новые темы и фоновые изображения. Невооруженным глазом не сразу заметишь, что версия на базе Slackware сменилась сборкой из исходных текстов. Это сокращает зависимости и повышает быстродействие системы.

Вот некоторые приложения, которые вы найдете в Puppy: *SeaMonkey*, *Pidgin*, *Sylpheed*, *PicPuz*, *Gnumeric*, *HomeBank*, *MTPaint*, *AbiWord*, *InkLite*, *Gtkam*, *Xine* и *OSMO*. Оригинальные инструменты настройки включают *Pburn* (прожиг CD/DVD), *Pwireless* (сканер беспроводных сетей) и *PupX* (настройка X-сервера). Другие любимые вами приложения можно установить при помощи *PETget*, менеджера пакетов Puppy, который также доработан наряду с новыми функциями этого релиза.



› Лучшие доводы за сборку из исходных текстов см. на стр. 42.

Puppy включает множество драйверов для беспроводных карт и модемов, которые не поддерживаются Linux напрямую. К тому же он умеет воспроизводить многие мультимедийные форматы, включая зашифрованные DVD. Он даже содержит несколько функций для ноутбуков, например, слежение за батареей.

Вы можете запустить Puppy с LiveCD, инсталлировать его на винчестер или загрузить с USB-диска. Варианты на USB или DVD/CD способны сохранять ваши персональные данные и восстанавливать их при каждой загрузке.

Puppy Linux развивается с 2002 г., но разработчики никогда не изменяли своей привязанности к легкому, быстрому и простому в использовании ПО.

www.puppylinux.org

Нарушения GPL опасны для всех

Недавно на нескольких сайтах прошли обсуждения о соответствии дистрибутивов Linux лицензии GPL. Охотник за ляпами и GPL-детектив Раду-Кристиан Фотеску [Radu-Cristian Fotescu] с http://beranger.org заявил, что большинство, если не все, обозреватели Linux не указывают, доступны ли исходные тексты. Признаю свою ошибку. В целях отслеживания ситуации, Раду помог мне начать таблицу, описывающую доступность исходных текстов. Если вы знаете другие, черкните письмецо.

Дистрибутив	Доступность исходных текстов
Zenwalk	Весьма частично для версии 5.0 и полное отсутствие для других.
Arch	Нет, но обещают предоставить в будущем.
Sam	Частично.
Nonux	Нет.
Slax	Только ядро.
Pardus	Запутанное собрание, в лучшем случае частично.
Puppy	Битая ссылка.
Damn Small	Только предложение записать за $7.
SimplyMepis	Продаются на DVD за $34.95 + доставка.
Wolvix	Лишь частично на медленном зеркале, предложение записать с оплатой доставки.
EasyS	Ранее не было, но теперь древо заполняется.

Хит-парад дистрибутивов

10 самых посещаемых страниц на DistroWatch.com со 2 по 30 мая 2008 (среднее число визитов в день).

	Дистрибутив	Число посещений	
1	Ubuntu	3,702	↔
2	OpenSUSE	1,765	↔
3	PCLinuxOS	1,404	↑
4	Fedora	1,379	↓
5	Mint	1,321	↑
6	Mandriva	931	↓
7	Debian	841	↔
8	Sabayon	746	↑
9	Damn Small Linux	673	↑
10	Dreamlinux	672	↔

» **DistroWatch.com** оценивает популярность дистрибутивов, основываясь на числе визитов на страницу каждого из них. Хотя это не дает представление о действительном числе установок, но показывает, какие дистрибутивы были более популярны за определенный промежуток времени. LXF

Linux на чем угодно

Управляйте своим миром, установив Linux на игровые консоли, настольные компьютеры, беспроводной маршрутизатор… да почти на все, был бы ЖК-экран.



Наш эксперт

Джон Брэндон – ветеран компьютерного движения с 20-летним стажем, использующий Linux с самого момента его появления, наряду с другими ОС (в том числе Lisa, OS/2, Mac и кое-чем еще). Он – бывший ИТ-директор, превратившийся в журналиста, и в этом его уникальность.

Известная латинская пословица гласит: «Кто повсюду, тот нигде». В компьютерном мире это не совсем так. Linux проникает глубже и глубже, управляя все большим количеством настольных и мобильных систем, донося музыку и видео до вашего дома через сети цифрового вещания и используясь как платформа разработчика в таких студиях высококлассных видеоэффектов, как Industrial Light & Magic, ранее тяготевшей к Windows и Mac. Но многие устройства еще не управляются Linux'ом, полагаясь вместо этого на проприетарные ОС. Для энергичного пользователя Linux нет такой вершины, которую он бы не смог покорить. Linux можно установить почти на все, что имеет процессор, монитор и немного свободного места. Иногда не нужно даже и монитора, если вы подключаетесь к устройству через управляемый Linux'ом ноутбук или скидываете ядро с USB-брелка. Оживленная Linux'ом машина откроет перед вами новый мир разделяемых ресурсов, анализа сети и домашней автоматизации.

Для каждого из описанных ниже случаев мы приведем не просто шаги, необходимые для установки: вам ведь захочется узнать, что можно сделать с обновленным устройством. Linux – не только повод похвастаться, он также придает аппаратуре дополнительные функции и возможности. PS3 станет домашним медиа-сервером, а маршрутизатор Linksys WRT54G превратится в сетевой анализатор, способный отражать атаки возможных хакеров. Это может быть интересным, побуждающим, но также и поучительным. Так что приготовьте ваши дистрибутивы (многие из них бывали у нас на LXFDVD) и приступайте к хакингу – пусть везде будет Linux!

Предупреждение!

Перед тем как взяться за дело, усвойте несколько простых правил. Во-первых, для таких дорогих устройств, как Sony PS3, перед установкой альтернативной системы убедитесь, что вы разобрались в гарантии и во всех рисках повредить приставку. Зарезервируйте свои данные: весьма вероятно, что инсталляция Linux затрет эти файлы. Даже если вам кажется, что безопасности данных ничего не угрожает (например, установка выполняется в новый раздел), все равно готовьтесь к сбою: иногда инсталляция проходит гладко, но при этом весь диск становится нерабочим, и его приходится переформатировать, приводя машину в исходное фабричное состояние. В некоторых редких случаях (например, для Sony PSP), установка Linux аннулирует гарантию, и вы обнаружите, что вернуть устройство в прежнее состояние очень сложно – то есть вы должны реально хотеть использовать его для Linux, а не для тех целей, для которых оно было разработано (т.е. игр).

Установка Linux на PSP

Сложность: Трудно Обратимость: Нет

Процедура простая и ясная; единственным камнем преткновения при установке Linux на PSP является нарушение гарантии (поскольку сначала вы должны взломать PSP OS), при сбое устройство может превратиться в «кирпич», а вернуть PSP в исходное состояние сложно. При моих тестах я обзавелся «кирпичом» вместо PSP, даже не успев ничего попробовать – видимо, из-за дефектной карты MemoryStick Duo. Карманные устройства более хрупки, чем ПК, и один-единственный ошибочный кусок кода может повредить крошечное (и часто основанное на флэш-памяти) внутреннее хранилище данных, сделав консоль неработоспособной.

Приведенные инструкции будут работать, только если вы сначала «раскурочите» вашу PSP. Мы не собираемся учить вас вскрывать батарею (это опасно), тем более что есть куда более простой путь. Поищите в сети пакет «pandora's battery»: он включает новую батарею PSP и карту MemoryStick Duo с необходимым самопальным ПО. Они стоят всего несколько долларов и избавят вас от головной боли, если вы не знаете, как припаивать проводки от батареи. Найдите на вашем PSP болты, открутите их, вытащите старую батарею, засуньте новую, затем верните крышку на место. Вставьте карту MemoryStick Duo и загрузите новое системное ПО.



Еще совет? Можете купить самопальную PSP на eBay или даже отослать ваш PSP на специальный сайт, где делают всю хирургию, чтобы устройство стало перепрошиваемым и могло запускать Linux. Конечно, если у вас уже есть старая PSP, которая не обновлялась до версий PSP OS 2 или 3, весьма возможно, что вы сможете установить самопальную ОС и прошить Linux безо всяких проблем.

PSP Linux — легкие шаги

Загрузите прошивку

Рекомендуем взять новую с иголочки карту MemoryStick Duo: старые сбойные носители для наших целей не годятся. Подключите кардридер к вашему ПК и вставьте в него карту MemoryStick Duo. (Вы не сможете подключить PSP прямо к Linux-машине без добавления специальных приложений или использования командной строки – это лишняя морока, а в Ubuntu Hardy Heron, похоже, вообще не работает.) Теперь скачайте самопальную прошивку для PSP. Мы использовали вариант от Dark Alex с www.dark-alex.org. Зайдите на сайт, щелкните на PSP, загрузите последний хак M33-3 и скопируйте всю папку UPDATE на карту MemoryStick. Также скачайте дистрибутив Linux для PSP с сайта Мо Джексона [Mo Jackson]: http://jacksonm88.googlepages.com/linuxonpsp.htm. После загрузки нужных дистрибутивных файлов и отката вашей PSP до подходящей версии системного ПО, вставьте карту в консоль.



1 Хакнем
Воспользуйтесь самопальной ОС от Dark Alex, позволяющей установить Linux.



2 Возьмем MemoryStick
Скачайте дистрибутив uClinux и сохраните его на карте MemoryStick Duo.



3 Примем на борт
Скопируйте самопальную ОС и дистрибутив Linux на карту, используя ваш ПК.

Запустим установщик

В XMB (XrossMediaBar, это ОС на PSP), войдите в меню Game и выберите карту MemoryStick Duo. Сначала запустите установку прошивки ОС M33, нажав на X; дождитесь окончания и перезагрузитесь. Если появятся сообщения об ошибке, вам, видимо, придется откатиться к предыдущей версии системного ПО Sony PSP. Установив прошивку, опять войдите в меню Game, выберите файл **uClinux** и нажмите X. Запустится установка uClinux. Дождитесь перезагрузки, и вы увидите командную строку. Теперь можно запускать приложения uClinux – например, клиенты обмена мгновенными сообщениями, браузеры, игры и видеоутилиты.



› **Запустив Linux на PSP, вы можете использовать интернет-пейджер, видеопроигрыватель и другие утилиты из-под uClinux.**

»

Маршрутизатор с Linux-начинкой

Сложность: Легко **Обратимость:** Да

А вот переделка с осязаемой практической целью. Установив Linux на маршрутизатор, вы сможете использовать устройство для анализа сети, расширить его функции за пределы обычной спецификации (добавкой мер безопасности или настройки учетных записей пользователей) или поэкспериментировать с новыми хаками, вроде потокового вещания на другой беспроводной гаджет в вашей квартире. По сути, вы идете в обход официальной прошивки и открываете дверь новым приложениям. Можно изменить настройки маршрутизатора так, чтобы конкретный Linux-драйвер лучше работал в Wi-Fi сети. Чтобы следовать нашим инструкциям, будет лучше иметь маршрутизатор точно такой же марки и модели, хотя базовые принципы одинаковы почти для всех устройств, для которых доступен установочный образ. (Вы можете произвести поиск Google для вашей модели – например, «D-Link DIR_655 Linux» – чтобы увидеть, доступна ли прошивка, затем скачать и установить ее, следуя руководству ниже.)

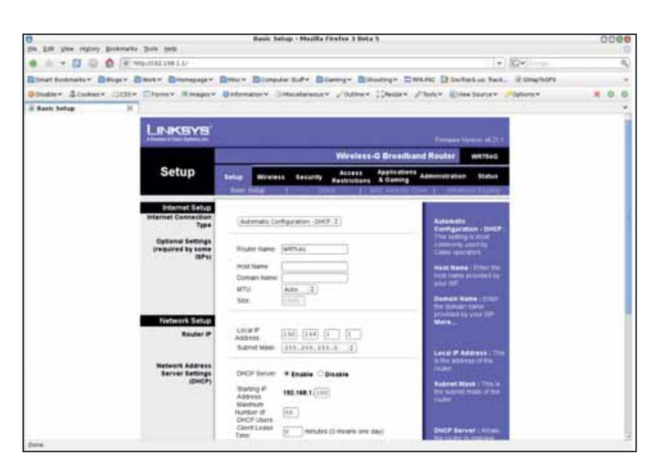
Мэтью Кэрон [Matthew Caron], соавтор Linux-хака для маршрутизатора WRT54G, советует также внимательно рассмотреть номер модели. У популярного WRT54G имеется совершенно новая исправленная разновидность под названием WRT54GL, предоставляющая для установки Linux немного больше памяти. Также доступна пара различных версий Linux. Мы использовали DD-WRT (www.dd-wrt.com), но есть и вторая, под названием OpenWRT (www.openwrt.org), она более гибко настраивается. Кэрон отметил, что для анализа сети часто используется OpenWRT: установив его, вы сможете подключить маршрутизатор к сетевому коммутатору, а затем применить его для скачивания данных и анализа трафика. Это дешевый способ произвести некоторую оптимизацию WAN, чтобы ускорить приложения, если вы видите неполадки в сети, или просто посмотреть, как работает коммутатор. Для данной установки мы использовали файл DD-WRT v24 и маршрутизатор Linksys WRT54G v2.

1 Найдите поддерживаемый маршрутизатор

Первым делом следует убедиться, что ваш маршрутизатор Linksys поддерживается. Зайдите на www.dd-wrt.com/wiki/index.php/Supported_Devices и посмотрите список доступных моделей.

› Подключитесь к маршрутизатору через проводное соединение.





› Дистрибутив Linux загружается так же, как официальная прошивка маршрутизатора.

2 Подключите маршрутизатор

Подключите ваш маршрутизатор прямо к ПК с помощью кабеля Ethernet. Не пользуйтесь беспроводной связью: возможен конфликт при инсталляции. Зато взять ИБП – хорошая идея, потому что сбой питания во время установки может сделать маршрутизатор практически бесполезным и потребовать ремонта оборудования – будьте осторожны!

3 Скачайте Linux

Теперь загрузите ту версию Linux, которую вы собираетесь использовать в вашем маршрутизаторе. Вы сможете найти их на www.dd-wrt.com/wiki/index.php/What_is_DD-WRT%3F#File_Versions. Есть варианты, специально разработанные для VoIP, VPN, плюс стандартная и мини-установки. Если вы не очень понимаете, что значат эти слова, берите Mini – впоследствии вы всегда сможете обновить ее до стандартной или любой другой редакции. Загляните под маршрутизатор, чтобы узнать версию и номер модели, например, WRT54G V2.

Конкретная прошивка, использованная нами для WRT54G v2, находится здесь: www.dd-wrt.com/dd-wrtv2/downloads/stable/dd-wrt.v24/Broadcom/Linksys/WRT54G_v2/dd-wrt.v24_mini_wrt54g.bin.

4 Сбросьте маршрутизатор

Сбросьте устройство, нажав на 30 секунд кнопку Reset при включенном питании. Это очистит все настройки и приготовит маршрутизатор для установки вчистую. Известны случаи, когда специфические настройки конфликтовали с DD-WRT.

5 Обновите прошивку

Зайдите на маршрутизатор, вписав его IP-адрес в адресной строке браузера вашего ПК. Если вы не знаете IP-адреса по умолчанию, загляните в руководство пользователя. Сделав это, щелкните во вкладке Administrator и выберите Firmware Upgrade. Нажмите Browse и выберите Bin-файл, скачанный вами ранее. Щелкните по Upgrade и подождите, пока загрузится прошивка.

6 Настройка

Маршрутизатор перезагрузится, и вы увидите интерфейс DD-WRT. Здесь можно настроить параметры QoS, отрегулировать ограничения доступа к сети, детально посмотреть память и информацию о сетевых пакетах, увидеть SSH-информацию, сконфигурировать DHCP, VPN или оптимизировать VoIP-маршрутизацию, а также задействовать режим моста, чтобы распространить беспроводной сигнал по всему дому, и даже настроить hotspot, чтобы к нему могли подключаться ваши гости. Отметим, что после установки DD_WRT, по умолчанию маршрутизатор будет использовать имя пользователя “root” с паролем “admin”, а не то, что написано в документации Linksys.

Двигатель сети

«Мне нравятся оборудование и сеть, и я много занимаюсь тем, чтобы интересные устройства делали замечательные вещи: например, подключаю WRT54G с измененной прошивкой к линии с VoIP-телефоном и смотрю, нельзя ли снять с него трафик и перехватить разговор, а также отслеживаю тех, кто стучится в двери моей сети, или настраиваю свою беспроводную сеть и пробую ее взломать. Тут мало чего нового, все это делалось и раньше, но так создается база для того, какие вопросы мне надо задавать и какие темы исследовать, чтобы расширить свои познания.»

Мы благодарим Мэтта Кэрона за содействие в подготовке материала.



› Мэтт Кэрон объясняет, почему он решил установить DD-WRT на свой маршрутизатор Linksys.Verfintem.

Установка Linux на Playstation 3

Сложность: Средняя Обратимость: Да

Установка Linux на Playstation 3 проста, и фактически она поддерживается и одобряется Sony. Похоже, компания рада разрешить конечным пользователям наращивать силу большой черной игровой консоли, в отличие от снобов из Microsoft (те заперли Xbox 360 так крепко, что пока никому не понятно, как водрузить на него Linux). Здесь же установка Linux – это вопрос выбора пункта меню и перезагрузки.

Единственная сложность – вам, быть может, придется выполнить инсталляцию несколько раз, пока все не заработает. PS3 случается капризничать на двух разных этапах: старте загрузчика и установке подходящей версии Fedora. Некоторые пользователи сообщали об отсутствии проблем на обеих стадиях, так что ваш опыт может варьироваться, но если вы споткнулись, просто начните все сначала.

1 Подготовьте PS3

PS3 надо подготовить к установке. Возьмите USB-брелок (подойдет любой объем, так как сначала мы будем копировать небольшие файлы) и вставьте его в ваш PS3. На PS3, используйте XMB (родную ОС) и зайдите в меню Game, затем выберите Saved Data Utility. Она позволит зарезервировать все данные, содержащиеся на PS3. Сохранив их, подключите контроллер PS3 в свободный USB-порт – если в ходе установки он потеряет связь с приставкой, ваши шанс обрести черный блестящий «кирпич» резко возрастут. Затем вытащите USB-накопитель и воткните его в ваш ПК. Выключите PS3, сэкономив тем самым на электричестве...

2 Загрузите и подготовьте файлы

Затем скачайте все файлы, которые понадобятся вам для установки. Загрузчик возьмите здесь: http://dl.qj.net/PS3-Linux-Addon-Disc-PlayStation-3/pg/12/fid/11308/catid/514. Установщик Linux (для официального процессора Sony PS3 Cell) – здесь: http://dl.qj.net/click.php?pid=111&fid=11328. И ISO-файл Linux – здесь: http://mirror.linux.duke.edu/pub/fedora/linux/core/5/ppc/iso/FC-5-ppc-DVD.iso.

В скачанном вами образе загрузчика вы увидите файл под именем **otheros.self** – скопируйте его в корень USB-накопителя. Чтобы открыть ISO-файл **Cell** (установщик Linux), используйте программу типа PeaZip, и отыщите там в папке **kboot** файл **otheros.bld**. Также перенесите его в корень, вытащите брелок и подключите его к PS3. Затем создайте DVD из ISO-файла Fedora Core 5. В программе для прожига DVD выберите наименьшую возможную скорость, например, 2X. Затем извлеките диск и вставьте его в PS3. Приставка загрузится в XMB.

3 Установите Linux

В PS3, перейдите в меню System > System Settings > Format Utility. Создайте новый раздел, выделив под Linux 10 ГБ. Вернитесь в меню System Settings и выберите опцию Install Other OS. В подсказке щелкните Yes. PS3 автоматически найдет и считает с USB-накопителя загрузчик kboot; подтвердите установку. Теперь снова ступайте в System Settings, выберите пункт Default System, поставьте Other OS и подтвердите это. PS3 перезагрузится и запустит ядро Linux. В приглашении kboot введите:

```
install-fc sda
```

Запустится установщик Fedora Core 5. Следуйте указаниям этого хорошо знакомого вам инсталлятора.

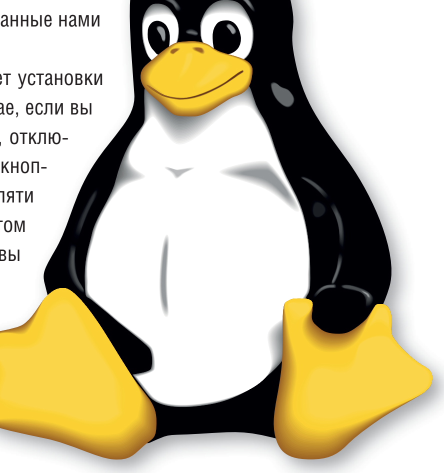


› Выберите пункт Other OS, запустите загрузчик Linux и укажите другую ОС, чтобы работать под Linux.

Внимание: Если ваш PS3 постоянно перезагружается или отказывается устанавливать Fedora, проверьте несколько вещей. На PS3, убедитесь, что используете самую последнюю версию XMB. Затем запишите диск с Fedora Core 5, используя наименьшую возможную скорость, и проверьте качество прожига. Наконец, попробуйте использовать версию Yellow Dog Fedora, лежащую в виде одного ISO-образа на www.terrasoftsolutions.com/products/ydl.

Fedora – явно один из самых распространенных дистрибутивов, и вы можете легко скачать любой нужный вам пакет, включая MythTV, программы для записи музыки, альтернативные браузеры и игры. В этом дистрибутиве вы можете использовать почти любые мышь и клавиатуру, в том числе протестированные нами модели Logitech G5.

Из всех устройств, проверенных нами на предмет установки Linux, PS3 оказалась самой снисходительной в случае, если вы решите вернуться к первоначальной ОС. Во-первых, отключите приставку, затем включите, но удерживайте кнопку Power (ту, что подсвечена красным) в течение пяти секунд, пока не услышите звуковой сигнал. При этом произойдет сброс PS3 и снова запустится XMB, и вы опять сможете играть в *Grand Theft Auto 4*!



› Сделайте резервные копии всех ваших файлов Grand Theft Auto, затем подсоедините контроллер через USB-кабель, чтобы он не потерял питание во время установки – это может принести несчастье!

Установка Linux на КПК

Сложность: Средняя **Обратимость:** Да

Современные наладонники, управляемые Palm OS или Windows Mobile, весьма мощные и, конечно, поддерживают множество разного и полезного ПО, но интерфейс обеих платформ начинает выглядеть поношенным (Microsoft планирует тщательный пересмотр Windows Mobile, но он, скорее всего, не появится раньше 2009 года). Инсталляция Linux на КПК даст вам больше контроля над тем, что вы можете сделать с карманным устройством, и поможет получить более развитую графику. После установки Linux, КПК становится больше похож на мини-компьютер. Понадобится всего несколько шагов, и выполнить их легко.

Мы протестировали дистрибутив Opie на Palm TX, но вы можете достать версии почти для любой модели Palm и использовать Familiar OS (familiar.handhelds.org) для КПК HP iPAQ (хотя последние модели, вроде iPAQ 110 Classic, не поддерживаются – да, мы пробовали!). Однако будьте осторожны, пытаясь обновить ПО на Palm Treo – можно нарваться на серьезные проблемы.

1 Скачайте приложения

Сначала загрузите нужное вам ПО. Opie для Palm TX доступен здесь: http://sourceforge.net/project/showfiles.php?group_id=155828. Вам также понадобится загрузчик, который можно скачать здесь: http://projects.linuxtogo.org/projects/palmtx/.

2 Подмонтируйте SD-карту

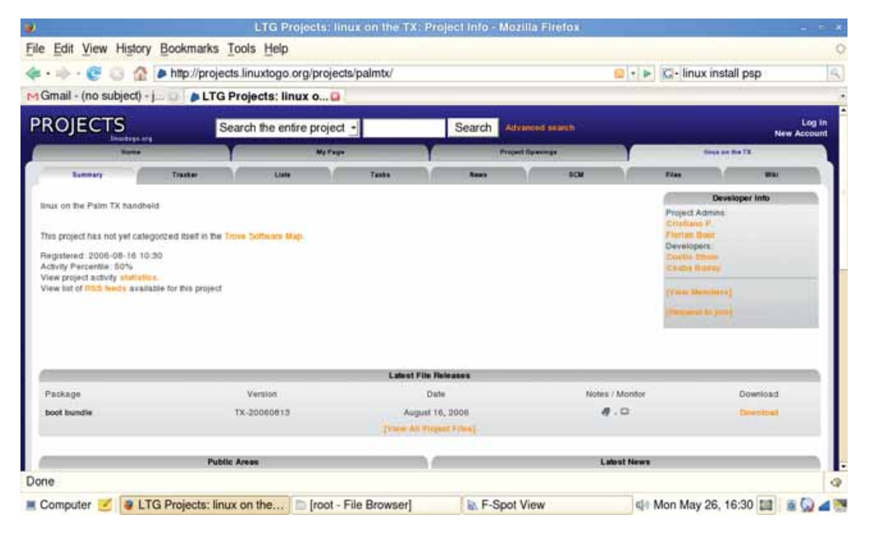
Palm TX использует карты SecureDigital (SD). Если у вашего ноутбука нет встроенного кардридера, потребуется внешний. Вставьте карту (неважно какого объема), затем скопируйте файл образа Opie и загрузчик в ее корень. Отмонтируйте носитель и вставьте его в TX.

3 Запустите загрузчик

Просмотрите память SD-карты на Palm TX и щелкните по файлу загрузчика Garux. Вас попросят подтвердить, что вы сделали резервные копии ваших файлов; если это так, то щелкните по "I did a backup", а если вам нужно сначала заархивировать файлы вашего Palm TX, выйдите из программы и проделайте это. Подтвердив, что резервные копии готовы, вы увидите экран загрузчика. Нажмите на Start Linux.

4 Запустите Linux

Вы увидите экран загрузки, который продержится где-то минут пять. В конце концов вы увидите экран-приглашение Opie. Ткните в него – вам предложат настроить стилус, прощелкав по специальным «мишеням». Следуйте экранным подсказкам и укажите язык, включите вкладку Docs, где сканируются и отображаются внешние носители, содержащие документы, а также задайте время. В итоге вы увидите главный экран Opie, откуда вы сможете запускать такие программы, как менеджер контактов и блокнот.



› Загрузчик Ganuz – это приложение, позволяющее запустить ядро Opie.



› Вставьте SD-карту и скопируйте в корневой каталог файлы для загрузки и запуска Linux.

› Запустите загрузчик, сделайте пару подтверждений и активируйте ядро Linux.

5 Назад, к Palm OS!

Установив Opie, вы сможете искать в Google программы для него и устанавливать их с использованием SD-карты. Похоже, что Opie не слишком хорошо поддерживается, а его сообщество, видимо, разбежалось. Интересно, что оно может возникнуть опять, поскольку все больше пользователей Palm осознают, что они могут реально расширить мощь своего устройства; становится все более очевидным и то, что Palm не сильно торопится создавать новую, более совершенную ОС для старых КПК. Работа с Linux на устройствах Palm дает вам желаемые функции, а также море полезных мобильных приложений – даже больше (и лучше), чем можно найти на сайтах вроде Handmark.com.

Возврат к первоначальной Palm OS прост: вы можете произвести полный аппаратный сброс, при этом будет загружена исходная ОС. Просто удерживайте кнопку Power, затем с помощью скрепки нажмите на кнопку Reset сзади устройства. Вы увидите логотип Palm; отпустите Reset и удерживайте Power, пока не появится второй логотип Palm вместе с индикатором выполнения внизу экрана; при этом ОС загрузится из ПЗУ. Когда вас, наконец, спросят про язык, который вы хотите использовать, знайте, что сброс произведен корректно и можно снова использовать Palm OS. Уфф!

Установка Linux на Nintendo DS

Сложность: **Средняя** **Обратимость:** **Нет**

В отличие от PSP, установка Linux на Nintendo DS потребует некоторых специальных приспособлений, которые позволят вам загружать на DS различное ПО. Гэрет Лэтти [Gareth Latty], житель Великобритании, является ключевым участником команды DS Linux, работающей над надежным дистрибутивом Linux для DS. Вы можете узнать о проекте больше на dslinux.org.

1 Настройка оборудования

Перед загрузкой дистрибутива вам понадобится кое-какое оборудование. Во-первых, это GBA Movie Player (GBAMP), устройство, работающее подобно старым GameShark для видеоконсолей; оно позволит вам загружать и запускать приложения на DS. Затем, понадобится карта CompactFlash (CF), которая вставляется в слот GBA Movie Player.

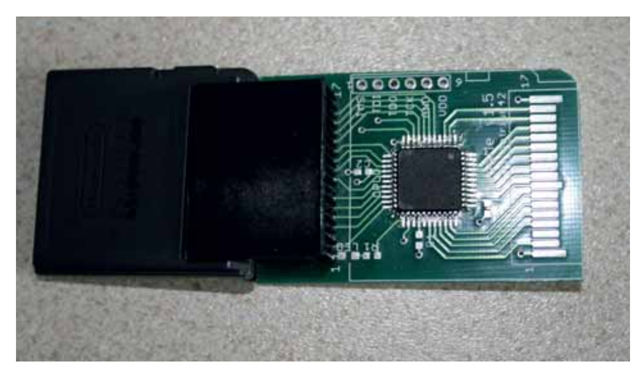
Вам также потребуется место для хранения файлов на DS, так как у него нет внутренней памяти, куда можно записать Linux. Для этого сгодится карта NoPass, типа Datel MaxMediaLauncher (www.datel.co.uk), позволяющая хранить файлы на DS. Некоторые карты NoPass также увеличивают ОЗУ DS, что приведет к более быстрой работе Linux. (Вы можете найти карты GBAMP и NoPass на eBay или Amazon.com, или магазинах б/у видеоигр, где продаются аксессуары.)

2 Подмонтируйте карту и загрузите Linux

Примонтируйте CF-карту под Linux, используя кардридер (или вставьте ее в слот ноутбука, если вы обладаете таковым). Скачайте дистрибутив Linux с http://kineox.free.fr/DS/dslinux-gbamp.tgz и разархивируйте его на смонтированную CF-карту. Скомандуйте:

```
tar -C /media/cf -zxvf dslinux-gbamp.tgz
```

Теперь поместите CF-карту в GBA-слот на GBAMP (тот, что побольше). Вставьте карты GBAMP и NoPass в ваш DS и загрузите консоль. Вы увидите командную строку, откуда сможете выбрать файл **dslinux.nds**, чтобы на Nintendo DS запустился Linux.



› **Карта NoPass содержит данные для установки DS Linux.**



› **Вставьте карту GBA Movie Player в большой слот на DS.**

3 DS в сети

Ну и что теперь можно делать? Новички могут перенести файлы с карты GBAMP на карту NoPass – просто скопируйте их из и в каталог /GBAMP; ваш DS превратится в карманное устройство хранения данных. DS Linux также содержит браузер под названием *retawq*, клиент обмена мгновенными сообщениями *tinyirc* и широкий спектр игр, например, *Sudoku*, *Boggle* и *Battleship*. Запустить любую игру можно из командной строки, просто введя ее название. Вы также можете использовать telnet, читать электронную почту и даже слушать интернет-радио.

Мы не будем здесь углубляться в детали, но для доступа к Интернету вам придется настроить беспроводную сеть для DS из DS Linux. То, как это сделать, изложено в видеофайле (на http://youtube.com/watch?v=xhQiqjt4nVA).

Если вы захотите снова играть в игры на DS, дорога будет трудной. Вы можете связаться с Nintendo насчет поддержки и ремонта, но они наверняка увидят, что вы установили Linux, и могут отказать вам, поскольку гарантия аннулируется. Некоторые пользователи сообщали, что им удавалось архивировать исходные файлы DS и загружать их заново, но у нас нет подтвержденных сведений о том, что это работает. Это дорога в один конец: нам не удалось переустановить операционную систему DS после развертывания Linux ни на оригинальной, ни на малой (DS Lite) версиях этого карманного игрового устройства.

Благодарности

Благодарим Гэрета Лэтти за его руководство при установке Linux на Nintendo DS. Дополнительная информация о нем – на сайте www.lattyware.co.uk.





› **Так ли страшен DS Linux? Страшно сказать, насколько страшен.**

Установка Linux на Xbox

Сложность: Трудно **Обратимость:** Нет

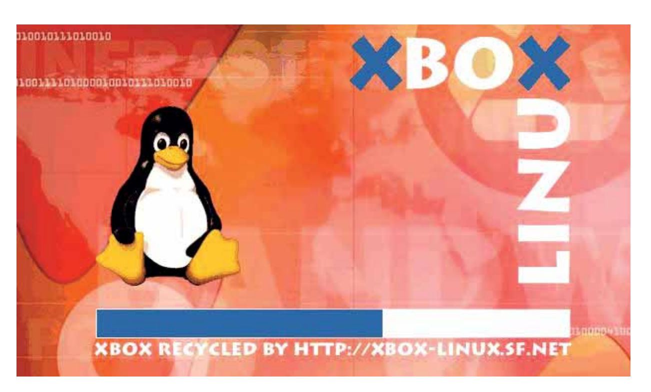
Это немного «старомодно» – ведь сейчас главной игровой консолью Microsoft является Xbox 360 – но загрузка Linux на первый Xbox (популярный около 2002 г.) имеет смысл: старая машинка, пылящаяся в углу, обретет новую жизнь. Кроме того, Xbox – все еще жизнеспособный ПК с процессором Celeron, видеокартой NVIDIA GeForce 3, DVD-приводом, звуковой системой Surround, портами USB, Ethernet, жестким диском на 10 ГБ (в большинстве систем) и 64 МБ ОЗУ. По сегодняшним понятиям это не бог весть что, но Xbox вполне можно превратить в дополнительный Linux-терминал. Для установки Linux я использовал старый пакет разработчика Xbox, и он работает как пчелка, если четко следовать пунктам и не отклоняться от накатанной тропы.

Внимание: Чтобы инсталлировать что-нибудь на Xbox, вам придется обойти почти герметичную защиту Microsoft: она препятствует установке любых приложений третьих лиц, в том числе и Linux. Вы можете взять простое устройство под названием Action Replay (**www.codejunkies.com**), которое позволит вам обращаться к жесткому диску. В сущности, это кабель USB-на-Xbox, используемый для передачи файлов. Вы также можете припаять обычный USB-кабель к шнуру контроллера Xbox, или даже прорезать отверстие в Xbox Memory Unit и добавить USB-адаптер таким способом, но эти варианты немного рискованны, если вы неправильно подберете провода. За подробностями по созданию собственного кабеля обратитесь на **xbox-linux.sourceforge.net**. Вам также понадобится экземпляр игры *MechAssault*, позволяющей установить Linux как эксплойт.

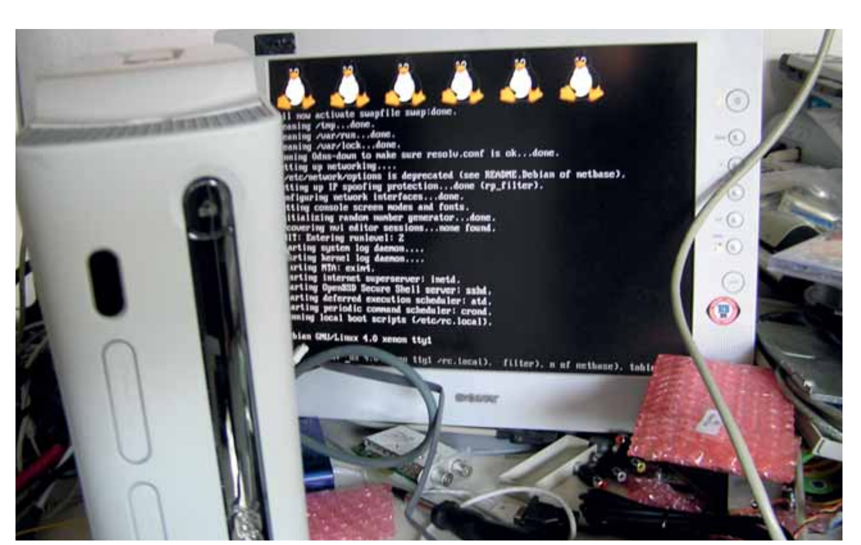




› Установка Linux на Xbox позволит вам играть почти во все игры для Linux на консоли, снятой с производства.



› Оживление Xbox неплохо и забавы ради, и для выгоды, но интересно и добиться большего от старого устройства.



› Станет ли Xbox 360 следующим, кто покорится Linux'у? Надеемся, да!

1 Соберите оснастку

Как и в случае с DS Linux, самая трудная часть установки Linux на Xbox – это сбор всего необходимого: кое-что уже не так часто встретишь. В моем случае я все перерыл в поисках Xbox Memory Unit и много думал, как скопировать файлы на диск Xbox. Это сложнее, чем кажется: в Xbox проприетарное USB-соединение, и мне пришлось расщепить стандартный USB-кабель и припаять его к шнуру Xbox – штука непростая, но она работает. Рекомендую воспользоваться Action Replay: он позволит подключить Memory Unit к вашему ПК, хотя это устаревшее (но легко отыскиваемое на eBay) устройство поставляется с программами только для Windows.

В любом случае, файлы, которые надо скопировать на Xbox, находятся здесь: **http://sourceforge.net/project/showfiles.php?group_id=54192**.

После загрузки Memory Unit, войдите в раздел Memory вашего Xbox. Выберите Memory Unit и файл Install Linux, затем Copy и Hard Disk. Повторите эти действия для файлов Emergency Linux и Restore Dashboard save game.

2 Загрузите Linux

Перенеся файлы на жесткий диск, нужно скопировать игру MechAssault для Xbox. Убедитесь, что это не версия Platinum Hits, а именно старый вариант – прокат из видеомагазина поможет вам в этом. Если вы купите совсем новую копию, скорее всего, ничего работать не будет. В игре выберите Campaign, затем Install Linux. Укажите файл Emergency Linux save game, нажмите A на контроллере, и вы увидите минимальную версию Linux. Затем можете перейти в командную строку и выполнить установку. Я рекомендую дистрибутив XDSL, созданный для Xbox и хорошо работающий на машинах с такими характеристиками. Он доступен здесь: **http://downloads.sourceforge.net/xbox-linux/xdsl-0.5.zip?modtime=1134953734&big_mirror=1**.

Эксцентричные варианты

Люди устанавливают Linux на странные штуки...

Linux на мотоцикле

Honda Ubuntu? Конечно, просто спросите Гарольда Симса из Сан Антонио, штат Техас.

«В связи с растущим интересом к внедрению Linux я подумал, что могу поделиться своим опытом. Я портировал ядро 2.6 на систему управления двигателем Marelli своего мотоцикла Ducati 1098, и просто счастлив, по сравнению со своей предыдущей системой. Хотя с моим предыдущим портом Vista меня ни разу не штрафовали за превышение скорости, постоянные аварии надоели. Сами понимаете, боковые панели Ducati очень дороги!»

Мы пришли в восторг от таких перспектив и связались с Гарольдом Симсом [Harold Sims]. Оказалось, что это шутка, которую мы обязаны были распознать, если бы прочли письмо повнимательнее, уже из-за одного подкопа под Vista. «Видимо, сказалась разница между бриттами и янки», – ответил Симс, когда мы спросили его, шутит ли он. «Я-то думал, шутка очевидна. Либо вы читаете и отвечаете на электронную почту в оперативном офисном стиле, с малым временем на обдумывание, либо редакторам требуется немного погулять в парке».

Оказалось, что Симс – ветеран-компьютерщик, и в мире Linux он «на всякое насмотрелся». В ходе переписки он объяснил подробнее свою шутку об «установке Linux на всем, что угодно». Он бывший сисадмин, работавший на машинах VAX Class Mini, был релиз-инженером рабочих станций SUN и UNIX и играл с HP-UX и Solaris еще в каменном веке. «Увидя ваш портрет в LXF, я подумал, что Джон Брэндон, наверное, стажер LXF, которого спустили с цепи повеселить редакторов», – объяснил Симс. «Я представил парня с деревянной ногой, которую откусил строчный принтер, когда он пытался поменять массивную красящую ленту. В любом случае, за последние пару лет я стал уделять больше внимания Linux'у, это как впадение в детство».



› Гарольд Симс, ты нас прокатил со своей шуточкой про Linux на мотоцикле...

Установка Linux на JuiceBox

Эксперт по инсталляциям Linux Дейв Андерс рассказал о своих подвигах

Дейв Андерс [Dave Anders] – это установщик-вундеркинд, и он оборудовал Linux'ом несколько карманных устройств, таких как Pixter, JuiceBox, TV Now и ZipIt. Он говорит, что его главная мотивация – обучение, так как он использовал эти дешевые гаджеты для экспериментов с мод-чипами, пайкой и установкой без особой опаски сломать что-либо ценное (ведь всегда можно найти замену). Другая причина – повышение квалификации: «Один из аспектов аппаратного хакинга – это замечательная платформа для обратного инжиниринга и отладки», говорит он. «В реальности, документации по оборудованию и выводам часто не хватает. Будь то из-за смены работы или просто из-за неустойчивости проекта, но разработчику случается переключаться на развитие существующего продукта».





› Даже эти крохотные устройства могут управляться вашей любимой ОС!

Установка Linux на планшетный ПК

ОК, возможно, этот будет тривиальнее остальных...

Еще один дерзкий проект – инсталляция Linux на планшетный ПК. Это куда сложнее, чем просто установить последний дистрибутив и использовать его как ноутбук. Вашей целью является разузнать, как заставить планшет Wacom поддерживать работу из-под Linux, чтобы вы могли использовать интерфейс пера. Для новичков сообщим, что большинство дистрибутивов не могут установиться, потому что не воспринимают конфигурацию оборудования. В моих тестах я пытался загрузить Fedora, Ubuntu, и SUSE 10, и всякий раз установщик слетал на полпути. Мне удалось поставить Ubuntu Studio со всеми программными дополнениями без всяких проблем. EmperorLinux (www.emperorlinux.com) работает со многими планшетными ПК, но в этом случае вам нужно отослать ваше устройство в компанию, чтобы они сделали всю работу за примерно 350 долларов США.



› Планшеты: хороши, если у вас красивый почерк.

X-фактор

Кейт Паккард трудится над *X Window System* более 20 лет, но его по-прежнему восхищает будущее графического фундамента Linux.



X Window System, известная также как X11 или просто *X*, является стандартной базовой графической подсистемой в Linux и других клонах Unix. На конференции Fosdem 2008 в Брюсселе мы спросили уважаемого разработчика *X*, Кейта Паккарда [Keith Packard], что ждет нас впереди и почему он до сих пор столь нежно привязан к графике…

Linux Format: Что вы делаете на Fosdem?

Кейт Паккард: Вот уже три года разработчики *X* встречаются на Fosdem ради множества разнообразных докладов и презентаций. Обычно мы вывозим разработчиков *X* на вечер в Брюссель. И, как обычно, нас сюда влекут пиво, шоколад и мидии.

LXF: И о чем ваш доклад?

КП: Последние пару лет я работаю в Intel и занимаюсь различными аспектами внутренней инфраструктуры *X*-сервера, чтобы соединить оборудование Intel с более современными графическими архитектурами, которые мы разрабатываем. Около года мы перерабатывали гра-



фический драйвер Intel для более эффективного использования оборудования и создания еще более крутых штук – как обычно, ради ярких демонстраций. Так что в этом году мой доклад посвящен тем изменениям в архитектуре, которые мы делаем, и тем, которые нам еще предстоит сделать.

LXF: Эти изменения касаются *X* в целом, или только драйвера Intel?

КП: Изменений много и на разных уровнях. Основная вещь, которую нам предстоит реализовать в последующие годы – перенести изрядную часть функциональности драйвера в ядро, чтобы обеспечить большую стабильность и дать людям больше возможностей для быстрого переключения между пользователями, загрузки, интеграции между [текстовой] консолью и *X Window System*. Среди крутых нововведений, которые мы собираемся предложить в будущем – синий экран с данными о сбое.

ПРО XFREE86:
«Я был бы счастлив снова работать с ними в X Window System.»

LXF: Переключиться на текстовый режим ANSI?

КП: Именно – одна из главных причин, по которой разработчики ядра и больших систем не могут работать с графическими интерфейсами, в том, что когда происходит сбой, невозможно вытащить информацию.

LXF: А как насчет *BulletProofX*, прокладывающей себе путь в Ubuntu и другие дистрибутивы? Что, конечная цель в том, чтобы *X* вообще не требовал файла конфигурации?

КП: Безусловно! И мы уже сейчас очень близки к этому. Вы можете запустить *X*, если у вас чипсет Intel или ATI, используя драйвер с открытым кодом, без файла конфигурации. Одно из ключевых требований для этого заключается в том, что «в общем правильная вещь» – не всегда то, чего хотят все. Во многом наша работа за последние несколько лет была нацелена на обеспечение возможности внесения изменений после старта системы, чтобы ничего не настраивать, пока не запустится *X*. Добавление новых устройств ввода, подключение нового монитора и изменение его настроек необходимы этой новой среде, свободной от файлов. Войдя в систему, вы можете сказать: «Хочу настроить мышь для работы левой рукой», или: «Мне нужен планшет», или: «Пусть монитор моего ноутбука будет слева от монитора компьютера» – и тому подобное.

LXF: Мы всегда удивлялись, почему *X* не мог благополучно вернуться к старому стандартному разрешению 640x480, вместо выдачи в текстовой консоли сообщения «Экран не обнаружен»...

КП: [Смеется] Да, мы пытаемся исправить множество подобных проблем – чтобы, если файл конфигурации неправильный, система вернулась бы к настройкам по умолчанию, а не выдавала сообщения об ошибках.

LXF: Или жгла монитор! Полагаю, с LCD такой проблемы нет?

КП: На самом деле, есть! Если вы неверно запрограммировали внутреннюю панель, вы можете легко повредить свой LCD. Интересно, что *X* раньше умудрялся портить ЭЛТ-мониторы, но сейчас у них хорошая защита. Если вы неверно запрограммировали работу с LCD, управление ими переходит на постоянный ток, а это может привести к закипанию жидкости внутри ЖК-ячеек.

LXF: А я-то надеялся, что обойдусь несколькими сгоревшими резисторами и испорченными чипами там и сям!

КП: Нет-нет, разрушения могут быть катастрофическими. Дабы обойти эту проблему, драйвер Intel, в частности, не позволяет программировать тайминги видео – эта опция отсутствует. Он использует те значения, которые, по сообщениям карты, являются правильными.

LXF: Вы участвуете в работе над Cairo и freedesktop.org?

КП: Я по-прежнему во многом отвечаю за freedesktop.org. Это – зонтичная организация для множества разных проектов: мы предлагаем услуги хостинга, списки рассылки и т.п. Мы предлагаем хостинг на бесплатной основе в обмен на свободные программы.

Что касается Cairo, я продолжаю просматривать архитектурные изменения, которые должны в него войти, и руковожу работой инженеров очень высокого уровня, которые не участвуют в повседневной деятельности. »

» LXF: Cairo может похвастаться несколькими весьма крупными достижениями – движок рендеринга *Firefox 3*, например, будет создаваться на нем…

КП: Приятно наблюдать такое единение. Когда мы начинали проект Cairo, его целью был только рендеринг в *X*. А месяца через три мы осознали, что это было неправильно. И вместо этого он стал кроссплатформенной и поддерживающей различные устройства вывода библиотекой рендеринга, так что приложения можно писать на Cairo, и они будут работать на Windows, Mac OS X и Linux. Они могут создавать PDF, или PostScript, или обращаться напрямую к драйверам GDI в Windows, что придает программам феноменальную гибкость. Вы можете использовать одно простое API, и ваше приложение будет работать где угодно.

LXF: А чем Cairo отличается от библиотеки Anti Grain Geometry?

КП: Есть другие библиотеки рендеринга, и многие из них сосредоточены на тех же абстракциях, что и Cairo. Основное, над чем мы работали в Cairo, было создание понятного и простого в использовании дизайна API, а также стремление к этой цели – кроссплатформенности и поддержке различных выходных устройств.

По-моему, Anti Grain не поддерживает печати. А для приложения, которое создает презентации на экране и использует то же самое API для бумажной копии или вывода PDF, неплохо иметь API, пригодный для обеих целей.

LXF: Темпы разработки не замедлились после ответвления от *XFree86*? [До 2004 года XFree86 был основным поставщиком *X* в Linux, но многие дистрибутивы перешли на другие ветви по причине лицензионных изменений.]

КП: На самом деле, разработка ускорилась – и весьма заметно. *X.org* отвечал за развитие последние четыре или пять лет, и мы затеяли серьезные реформы оконной подсистемы. Конечно, у нас появились новые примитивы рендеринга, которые в настоящее время весьма эффективно использует Cairo, и мы говорим о серьезных изменениях функциональности вывода для мониторов и новшествах для планшетов и клавиатур «горячего подключения». Теперь мы говорим о перемещении всех процессов настройки и инициализации в ядро.

Инженеры Red Hat трудятся над получением более чистой системы загрузки, чтобы можно было загрузиться в консоль фрейм-буфера и мягко перейти в среду *X*, а не проходить через черный экран со вспышками. Ubuntu работает над такой же функциональностью. И еще мы работаем над тем, чтобы вы могли видеть сообщения о проблемах в ядре на экране! Конечно, такое никто не хочет видеть никогда, и за это постоянно потешаются над Windows; и все же, чего хотят администраторы крупных систем? Они хотят использовать GUI для настройки и управления системой, и чтобы при этом на случай отказа интерфейс оставался текстовым.

LXF: Верно: в такой ситуации *X* просто виснет, и невозможно понять, что вообще происходит.

КП: Точно, и поэтому-то мы и ждем с нетерпением, чтобы сообщения о сбоях появлялись на экранах!

LXF: Мне кажется, для вас есть *X* и есть ядро, и они стоят особняком друг от друга – вы разделяете их жирной чертой…

КП: Вообще-то черта не такая жирная, как нам бы хотелось. Проблема была в том, что *X* реализовывал значительную часть функций ядра в оконной системе – то есть в пространстве пользователя. В частности, у нас был код управления шиной PCI для перемещения видеокарты на случай, если ядро отобразит ее неверно. А для всего видео нам приходилось отображать регистры управления оборудованием в пространстве пользователя – в том числе и I/O-порты системы. А ведь это – сущее безумие! Так что приходилось запускать *X* от имени root, а одна из главных целей – запускать ее от имени пользователя без привилегий.

LXF: И когда это будет? Работы, наверное, масса?

КП: Ну, на самом деле мы уже очень близки к этому. Мы надеемся скоро предъявить демонстрационный пример. Работы не так уж много – код установки режимов уже в ядре, а у нас есть работающий образец X-сервера для фрейм-буфера. У нас уже были демо-показы X-серверов, использующих инфраструктуру ускорения ядра; осталось только обсудить окончательный вид интерфейса и собрать его. Это очень чистая работа – наконец-то мы знаем, что делаем [*смеется*]!

LXF: Итак, вы хотите впихнуть побольше кода в ядро – а не получится ли у *X* некий уклон в Linux?

КП: Вовсе нет; API ядра, которые мы разрабатываем, являются кроссплатформенными, и мы рассчитываем использовать те же самые интерфейсы в Solaris и BSD. У нас уже есть компоненты ядра этих систем для нашей 3D-среды, названной Direct Rendering Infrastructure, так что мы будем просто применять их. Мы не хотим ограничивать количество ядер, на которых мы можем работать, наша цель – просто решить основные проблемы.

> ПРО НАДЕЖНОСТЬ:
> «X раньше мог просто жечь ЭЛТ-мониторы, но сейчас они отлично защищены».

LXF: Какие отношения у вас сейчас с парнями из XFree86? Продолжаете ли вы делиться кодом?

КП: Я был бы счастлив снова работать с ними над *X Window System*. Мы сотрудничаем по проблемам безопасности, чтобы исправления могли мигрировать в их систему. Но, помимо этого, совместной работы у нас нет.

LXF: Как вы представляете себе взаимоотношения между *X.org*, *XGL* и *AIGLX*? Мы можем говорить о DRI и 3D, но все пользователи ПК дружно скажут: «О-о-о, колышущиеся окна!»

КП: Именно, и это вовсе не плохо. Колышущиеся окна, которые мы используем сейчас, обычно строятся поверх *AIGLX* и *XGL*. *XGL* – интересный эксперимент, и он продемонстрировал часть инфраструктуры, нужной нам для плотной интеграции 2D и 3D в систему. Сейчас *XGL* не слишком активно разрабатывается – я не знаю, планируют ли дистрибутивы его долгосрочную поддержку.

Вся функциональность *XGL*, по сути, сводилась к ускорению и непрямому *GLX*, позволяющему *X*-серверу задействовать API GL при наличии подходящих драйверов. Именно это предоставляет *AIGLX*, но он предлагает данную функциональность без необходимости серьезно изменять остальную систему, так что мы смогли использовать более инкрементный подход.

> ПРО СПОР KDE И GNOME:
> «Мне они не кажутся сильно разными: оба они — двумерные.»

LXF: Вы используете *Compiz*?

КП: Когда как. Я по большей части работаю без композитного менеджера, потому что мои 2D- и 3D-драйверы хронически находятся в стадии изменения. У меня есть группа QA в Intel, тестирующая разработку драйверов и изменения GL, которые мы производим, и они тестируют *Compiz* и 3D-игры, чтобы убедиться, что все работает. Но мой повседневный рабочий стол очень прост.

LXF: Значит, это не KDE и не Gnome?

КП: Вы будете смеяться, но до 2001 года я использовал *Twm*! А потом переключился на KDE, и он мне очень нравился, но недавно я перешел на Gnome. Мне они не кажутся слишком разными; оба они – двумерные. Я запускаю в них *Emacs* – чего еще нужно?

LXF: Линус Торвальдс недавно говорил, что ядро в какой-то мере «завершено», и самое интересное творится в пространстве пользователя. Вам не кажется, что *X* находится в подобной ситуации? Он в основном закончен, и все интересное происходит исключительно на стороне приложений?

КП: Я уже 22 года работаю над *X*, и продолжаю надеяться, что когда-нибудь он будет закончен! Ведь оборудование продолжает меняться, опыт пользователей продолжает меняться, и создатели приложений все так же жаждут новых разработок. Так что и требования к базовой системе продолжают меняться – отчасти даже быстрее, чем требования к ядру.

Однако, как мы знаем, скорость изменений ядра возрастает с каждым новым релизом. Хотя Линус и сказал, что оно стабилизировалось и работа выполнена, всегда есть много интересного, что можно попробовать, даже если в итоге это и не приведет к успеху. И при вовлечении в процесс новых людей появляются и новые идеи.

LXF: На своем ноутбуке я постоянно мучаюсь из-за необходимости иметь инструмент *915resolution*, чтобы получить полноэкранное изображение. Это планируется изменить в драйвере Intel?

КП: Он вам больше не понадобится – он теперь встроенный. Программа манипулировала таблицами BIOS, чтобы создать альтернативную запись для родного разрешения вашего экрана, поскольку сам BIOS был неспособен сгенерировать ее. В таблицах BIOS имеется фиксированное количество слотов, и они обеспечивают фиксированные разрешения.

Windows не использует BIOS для программирования видеокарты, и изготовители ноутбуков иногда не меняют таблицу – не вносят в нее шесть чисел. Поэтому инструмент *915resolution* фактически редактирует таблицы BIOS и замещает одно из разрешений – то, что используется наименее вероятно, например, 800 x 600. Теперь, когда драйвер Intel программирует оборудование сам, как драйвер Windows, мы получили возможность задействовать любой нужный нам режим. Это дает нам доступ к картам с ТВ-тюнером, с выходом HDMI и DVI – все теперь имеет полную поддержку.

LXF: Intel владеет немалой долей рынка графических чипов, и у ATI там тоже неплохая часть, так что все давление приходится на Nvidia, вам так не кажется?

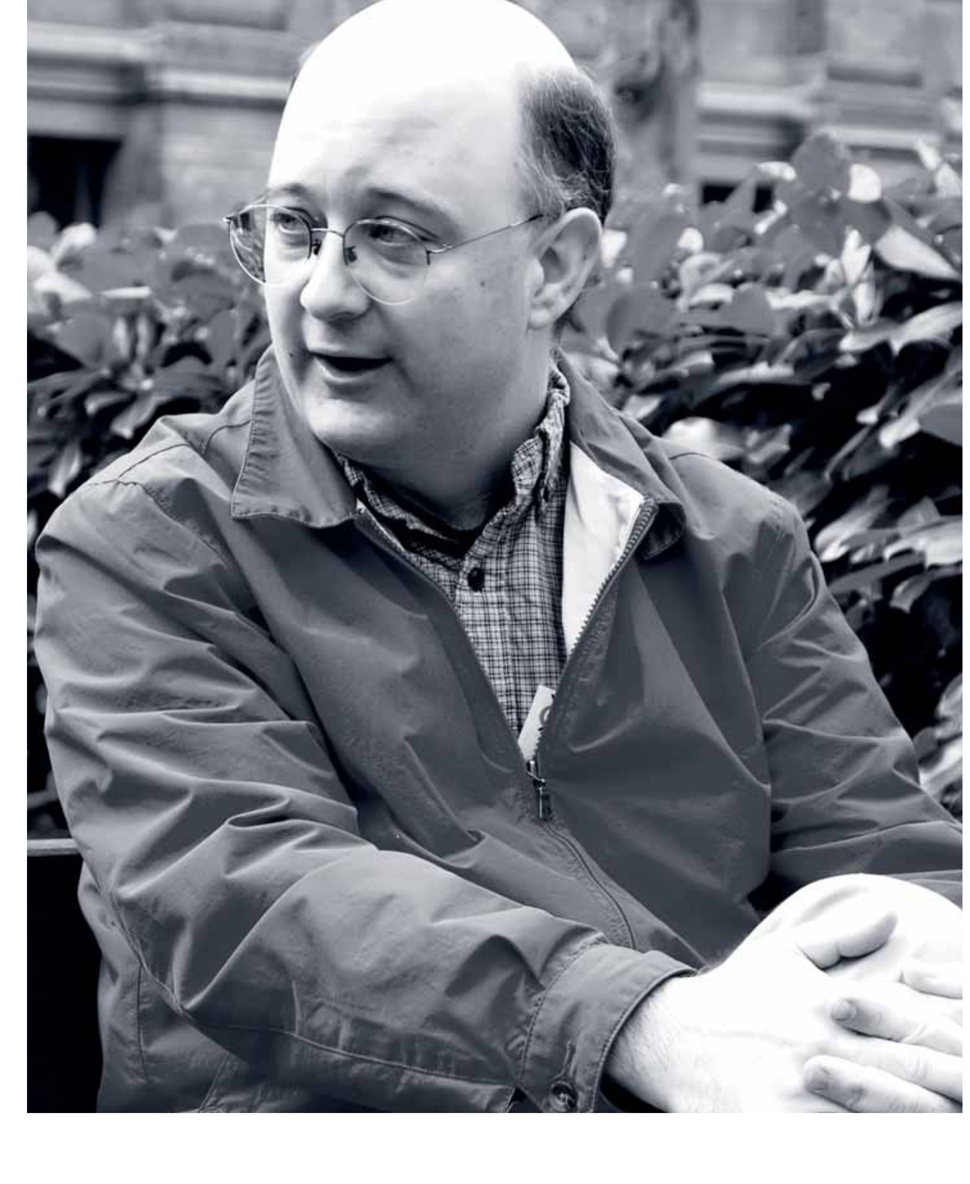
КП: Раз я ношу логотип *X.org*, мне бы, конечно, хотелось, чтобы они предлагали драйверы с открытым кодом, чтобы удовлетворить пожелания многих своих клиентов, но у них есть свои планы по разработке и маркетингу, и я не в курсе, что они планируют.

LXF: При загрузке драйвера с закрытым кодом ядро Linux ставит флаг, чтобы из отчета об ошибке разработчики знали: причина проблем, возможно, кроется в закрытом модуле. Для *X* планируется нечто подобное?

КП: Пока мы не видим в этом необходимости. Мы находимся в пространстве пользователя, а большая часть драйверов – в ядре. Если у вас есть видеодрайвер с закрытым кодом, значит, есть и компонент ядра с закрытым кодом.

Так что значительная часть оконной системы зависит от видеодрайвера; это – не ядро, где у вас может быть драйвер SCSI, сетевой драйвер, драйвер Wi-Fi, драйвер клавиатуры… а, и еще драйвер Nvidia. Вы не принимаете сообщения об ошибках, если в ядро был загружен драйвер Nvidia. Это все сильно отличается от практики *X*, когда у вас есть драйвер клавиатуры, драйвер мыши и огромный, всеобъемлющий видеодрайвер. И если вы получаете сообщение об ошибке от человека, который использует только-двоичный драйвер, или драйвер, отстающий на 47 ревизий (хотя и открытый), нам это неважно – мы вам просто скажем: «Используйте последнюю версию драйвера с открытым кодом, и сообщите нам, если в нем будет ошибка».

ОБ ОЖИДАЕМЫХ ФУНКЦИЯХ:
«В работе новые функции ввода для горячего подключения планшетов.»



LXF: Потратив столько времени на *X*, вы никогда не испытывали желания заняться чем-то другим?

КП: А я получаю ничуть не меньше удовольствия, продолжая заниматься оконной системой и созданием графики на 2D-уровне. И что особо приятно, моя задача постоянно меняется, каждый год. Двадцать лет назад мы были счастливы, когда на экран выводилось цветное изображение!

LXF: И, наконец, есть ли в *X* какие-нибудь «пасхальные яйца»? Например, комбинации клавиш, выводящие на экран изображение кролика, или что-то еще в том же роде?

КП: А что, было бы круто! В 1993 я работал над проектом X-терминала. Помните классический хранитель экрана *X*, в котором по экрану перемещалась плоская буква X? Мы сделали его так, что через каждые 300 или 400 изображений он выводил картинку Джо «Боба» Доббса [Joe “Bob” Dobbs], лидера Церкви Субгениев [псевдорелигиозная группа, пародирующая всяческие культы, – *прим. ред.*]. Мы отправили его некоторым нашим клиентам! [Смеется] У нас даже были звонки в службу технической поддержки с вопросом: «Это что еще такое?!» Они думали, их компьютер заражен каким-то жутким вирусом! LXF

ALSA: крутые советы и трюки

Дэниел Джеймс углубляется в звуковую подсистему вашего компьютера, чтобы извлечь лучшее из ALSA.



Наш эксперт

Дэниел Джеймс один из сооснователей дистрибутива 64 Studio. Он обожает выявлять и устранять проблемы с *ALSA* по неполным и туманным описаниям.

Архитектура *Advanced Linux Sound Architecture* **является стандартом в ядре 2.6, и в большинстве дистрибутивов есть автоматические средства конфигурации, выполняющие основную настройку. Поэтому вы, как минимум, слышите мелодию при загрузке рабочего стола. Но** *ALSA* **умеет гораздо больше.**

Почти в каждой настольной системе доступен простой микшер, возможно, в виде апплета регулятора громкости в лотке рядом с часами. Примерно так же оформлены элементы управления звуковым драйвером в Windows. Он очень удобен, когда во время разговора по телефону нужно быстро сделать потише CD, который вы слушаете, но не может согласовать уровни звука с различных источников.

Чтобы сделать это, откройте окно микшера, доступное в Gnome по щелчку правой кнопкой мыши на апплете регулятора громкости. (Если у вас не установлены *X.org* и рабочий стол, можно запустить аналогичную консольную утилиту командой *alsamixer*.)

Смикшируйте это

В окне микшера Gnome под каждым регулятором находится иконка «отключить» [mute], а у стереосигнала есть также «объединить» [chain], связывающая уровни левого и правого каналов. Кроме регулятора основного стереоканала, есть и регулятор стереоканала PCM, управляющий большей частью программного звука. Исключение из этого правила – системный звуковой сигнал: он генерируется независимо от звуковой карты аппаратным зуммером на материнской плате. В моем микшере есть регулятор PC Speaker, но он никак не влияет на громкость системного звукового сигнала. Вы также можете заметить регуляторы входных каналов: стереовход Line In или моно-регулятор для гнезда микрофона.

Регулятор ‘CD’ может ввести в заблуждение, потому что обычно относится к аналоговому выходу привода CD-ROM. Его нужно



› **На вкладке Patchbay/Router в окне** ***Envy24control*** **можно управлять физическими выходами звуковой карты.**

соединить тонким кабелем с четырехконтактным разъемом с аналоговым входом аудиокарты; полученный звук через цифровой кабель передается на современные приложения типа Sound Juicer в Gnome. Результатом будет не только лучшее качество звучания, но и «аналоговая дыра», которая так расстраивает производителей ПО с DRM. Если к вашему оптическому приводу не подключен аналоговый кабель, для управления уровнем звука CD придется использовать регулятор PCM.

Окна микшера выглядят незатейливыми, но с их помощью можно регулировать большинство параметров стандартных звуковых карт. Для доступа ко всем регуляторам микшера в Gnome воспользуйтесь диалоговым окном Edit > Preferences.

У обладателей устройств 5.1 должны присутствовать регуляторы объемного звучания или канала низкочастотных эффектов LFE (Low Frequency Effects) для сабвуфера. Этот канал – «точка-один» в системе 5.1, поскольку он не формирует полный диапазон частот, в отличие от других пяти. Также можно найти регуляторы каналов IEC958, которые относятся к гнездам цифрового аудио. Чаще всего это S/PDIF (Sony/Philips Digital Interconnect Format), и для передачи всех каналов 5.1 обычно требуется один кабель с коннектором RCA («тюльпан»). Реже можно встретить оптический аудиовход (световод) с разъемом с небольшой заслонкой или противопыльной крышкой, требующим оптоволоконного соединения Toslink (Toshiba link). Общее название таких гнезд – EIAJ Optical, по названию торговой организации Electronic Industries Association of Japan (Ассоциация электронной промышленности Японии).

Dolby Digital

На современных звуковых картах или материнских платах часто присутствует выход IEC958 для подключения усилителя домашнего кинотеатра, но нет цифрового аудиовхода. Тем не менее, если есть совместимый усилитель и в *MPlayer* или *Xine* установлена опция «прогон AC3» (AC3 pass-through), этот выход очень удобен для воспроизведения звука в стандарте Dolby Digital при проигрывании DVD-диска. Цифровой аудиовыход также рекомендуется использовать с *MythTV*.

С подъемом по ценовому диапазону у звуковых карт лучшего качества появляются входы, отсутствующие на потребительских устройствах или встроенных чипсетах: например, многоканальные оптические входы ADAT. (Аббревиатура ссылается на цифровой интерфейс, изначально использовавшийся в устройствах Alesis Digital Audio Tape.) На более дорогих аудиоплатах также могут присутствовать возможности измерений и маршрутизации, недоступные средствами стандартных микшеров *ALSA*, поскольку они специфичны для конкретных чипсетов. К счастью, для трех популярных моделей есть особые микшеры *ALSA*, каждый с графическим интерфейсом. Это:

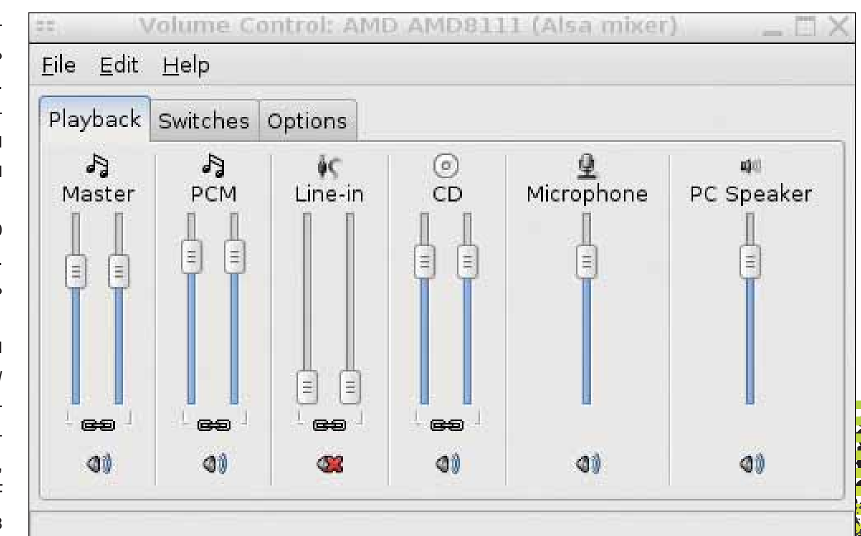
» **Envy24control** Поддерживает большинство звуковых карт PCI M-Audio Delta, Digigram VX442, Terratec DMX 6Fire – как Terratec EWS88, так и EWX 24/96, и Hoontech Soundtrack Audio DSP 24.

» **Echomixer** Предназначен для устройств, произведенных Echo Audio, включая PCMCIA-карты Indigo, MIDI-устройство Mia и PCI-карты Gina 3G и Layla.

» **Hdspmixer** Поддерживает звуковые интерфейсы RME HDSP, включая карты Hammerfall DSP 9632 и 9652 PCI.

Микшер особого назначения

Более продвинутые микшеры не только предоставляют возможности аппаратного измерения и маршрутизации, но и содержат функции управления, включающие блокировку частоты дискретизации и бит «защиты от копирования» SCMS, который есть в некоторых реализациях S/PDIF (ранняя форма DRM). В не-мультимедиа дистрибутивах эти микшеры не устанавливаются по умолчанию, но, как правило, для них есть готовые пакеты. Если вы загрузите исходники *ALSA* с сайта www.alsa-project.org, то найдете специализированные микшеры в пакете **alsa-tools**. В некоторых дистрибутивах Linux, включая Debian, этот пакет разделяется на консольные и графические приложения, и в производных Debian упомянутые выше графические микшеры находятся в пакете **alsa-tools-gui**.



› **Многие возможности микшера Gnome по умолчанию скрыты. Чтобы увидеть больше каналов, воспользуйтесь окном Preferences.**

Envy значит «завидовать»

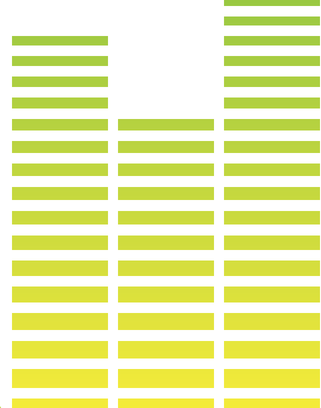
Благодаря популярности звуковых карт среднего класса типа M-Audio Audiophile 24/96, из трех микшеров вам скорее всего встретится *Envy24control*. Эта PCI-карта – хороший выбор для подключения hi-fi-систем или домашних кинотеатров, так как она позволяет управлять аналоговым уровнем на фоно-гнездах и поддерживает S/PDIF на вход и на выход. Устройство опробовано и протестировано под *ALSA*, а значит, это разумный выбор для профессиональных проектов, когда нужна только пара аналоговых каналов с несбалансированными входами/выходами – они прекрасно работают, пока соединяющие аппаратуру кабели короткие. Если вам требуются провода длиной в несколько метров, подойдут сбалансированные кабели, обычно с трехконтактными разъемами XLR или четверть-дюймовыми штекерами TRS (tip, ring, sleeve – наконечник, кольцо, муфта), поскольку сбалансированные системы способны подавлять электрический шум. Если необходимо восемь каналов со сбалансированными входами/выходами, можно взять M Audio Delta 1010, которая использует все тот же бывалый драйвер *ALSA*.

> «Envy24control подключится к первой поддерживаемой звуковой карте в системе.»

После установки подходящей звуковую карты, *Envy24control* можно запускать с терминала. В противном случае он откажется работать, выдав сообщение «Не найдено карт ICE1712» ('No ICE1712 cards found'). (Ссылка на ICE1712 отражает номер модели оригинального чипсета Envy24, а также название соответствующего модуля *ALSA*). У счастливчиков после установки пакета *alsa-tools* или *alsa-tools-gui* в меню рабочего стола появится пункт меню для *Envy24control*.

После старта *Envy24control* подключится к первой поддерживаемой звуковой карте, обнаруженной в системе. Адрес и номер прерывания IRQ для нее появятся в заголовке окна приложения, например, '0xd400, IRQ 10'. Если в системе несколько поддерживаемых звуковых карт, можно управлять каждой из отдельного приложения *Envy24control*, используя опцию **-c** с номером карты в командной строке. На man-странице *Envy24control* можно найти описание других полезных аргументов *Envy24control*.

В левой части окна *Envy24control* есть измеритель для «цифрового микшера» ('Digital Mixer'), показывающий сумму уровней внутренних»



» сигналов. На двух первых вкладках справа отображаются уровни входного и выходного сигнала соответственно. Как и в обычных микшерах *ALSA*, имеются кнопки отключения звука (mute) и набор стереокнопок. Есть и более необычная возможность. Уровень каждого входного сигнала, поступающего в левый и правый каналы цифрового микшера, можно контролировать отдельно, что напоминает управление панорамированием. Обычно для стерео первый вход поступает только в левый канал, а второй – только в правый.

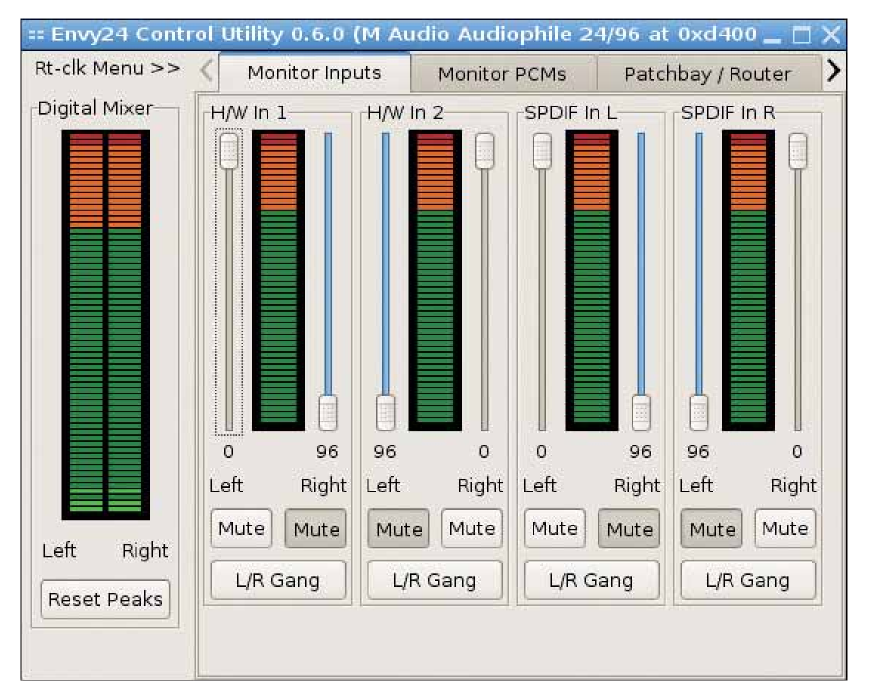
Patchbay/Router

Следущая вкладка *Envy24control* называется Patchbay/Router [Коммутационная панель/Маршрутизатор]. В стандартных потребительских звуковых картах внутренняя маршрутизация звука прошита аппаратно, но в более дорогих моделях появляется большая гибкость. Например, при желании можно связать аналоговый вход непосредственно с выходом S/PDIF.

Далее идет вкладка Settings [Настройки]. Наверное, самая важная настройка здесь – Master Clock [Главный генератор тактовых импульсов]. Можно выбрать стандартную частоту дискретизации внутреннего генератора тактовых импульсов звуковой карты от 22050 Гц до 96000 Гц или заставить карту работать на частоте внешнего устройства S/PDIF. Также можно зафиксировать выбранное значение частоты дискретизации или разрешить ее программный сброс. Эта настройка сначала может сбить с толку, если попытаться записать звук на частоте дискретизации 96000 Гц, потому что часто карта по умолчанию фиксирует 48000 Гц. На этой вкладке также можно задать параметры выхода S/PDIF.

Для звуковых карт 24/96 следующая вкладка носит название Analogue Volume [Аналоговый уровень], но она есть не у всех карт Envy24. На ней можно настроить параметры входного сигнала, поступающего на аналого-цифровой преобразователь (АЦП), и выходного сигнала с цифро-аналогового преобразователя (ЦАП), чтобы совместить уровни сигналов с внешним оборудованием и предотвратить искажения. На других интерфейсах, включая Delta 1010, для этого предусмотрены аппаратные переключатели рядом с каждым штекером. Это необходимо потому, что уровни аналогового сигнала не одинаковы во всей

«Иногда устройства типа winmodem — просто дрянные звуковые карты.»



› С помощью *Envy24control* можно панорамировать аппаратные входы для каждого источника.



› Аппаратные настройки позволяют зафиксировать частоту дискретизации.

звуковой аппаратуре. В профессиональных системах используется стандарт +4dBu, а в потребительских hi-fi-системах принят стандарт -10dBV. Все, что нужно о них знать – это что один громче другого; и если сигнал искажен (слишком громкий) или слишком тихий, причина может крыться в несоответствии уровней сигнала.

На предпоследней вкладке Profiles [Профили] можно сохранить и восстановить настройки микшера в *ALSA*. Наконец, на вкладке About [О программе] просто упоминается разработчик *ALSA* и ответственный за звук в ядре Linux Ярослав Кысела [Jaroslav Kysela], создавший *Envy24control*.

Изгоняем winmodem

В современных рабочих столах Linux устройств *ALSA* обычно два или три. В большинстве ПК есть встроенная звуковая карта или устройство PCI, но у вас также могут быть включены USB-колонки или VoIP-гарнитура. И то, и другое взаимодействует с *ALSA* независимо от встроенного звукового оборудования. Также может присутствовать карта MPEG-декодера или MIDI-устройство USB с собственным интерфейсом *ALSA*. Вдобавок, некоторые устройства типа winmodem на самом деле представляют собой дешевые и дрянные звуковые карты, в которых «бормотание» модема эмулируется программно. Вы экономите несколько рублей/долларов на микросхемах и повышаете загрузку процессора только ради ностальгии по dial-up.

Это по большому счету ничего не значит, не считая того, что в некоторых приложениях поддерживается только одно звуковое устройство, а другие все еще ожидают найти его в **/dev/dsp/** – устройство для первой звуковой карты из наследства Open Sound System (OSS), которого уже много лет нет в основном ядре Linux. В *ALSA* первому устройству дается нулевой индекс. На него ссылаются как на **hw:0**. К счастью, *ALSA* хорошо эмулирует OSS, но если звуковых устройство несколько, то только одно из них будет **/dev/dsp/** – остальные будут эмулироваться как **/dev/dsp1/**, **/dev/dsp2/** и т.д.

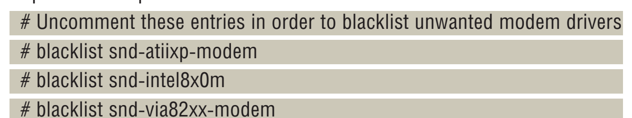
Вы вряд ли захотите, чтобы звонок по VoIP проходил через ТВ-приставку, а система hi-fi пищала каждый раз, когда пришло новое письмо. Поэтому нужно научиться определять индекс *ALSA* каждого звукового устройства и убедиться, что он не изменяется после перезагрузки.

Во-первых, избавимся от драйвера winmodem, если он не используется; в эпоху широкополосных соединений и Wi-Fi это скорее всего так и есть. Откройте терминал и проверьте, загружен ли любой драйвер winmodem с помощью команды:

```
$ lsmod|grep snd
```

Поищите модули *snd_atiixp_modem*, *snd_intel8x0m* или *snd_via82xx_modem* для чипсетов ATI, Intel и VIA соответственно. Модуль от Intel также поддерживает чипсеты winmodem от SIS и Nvidia. Будьте внимательны, если у вас встроенная звуковая карта и загружен модуль *snd_intel8x0*: это обычное звуковое устройство, а не модем, но имя модуля отличается лишь на одну букву.

ALSA предоставляет механизм, предотвращающий загрузку драйверов winmodem; в основанных на Debian дистрибутивах Linux он управляется файлом **/etc/modprobe.d/alsabase-blacklist**; в других системах расположение может немного отличаться. В файле должны быть примерно такие строки:



```
# Uncomment these entries in order to blacklist unwanted modem drivers
# blacklist snd-atiixp-modem
# blacklist snd-intel8x0m
# blacklist snd-via82xx-modem
```

Естественно, редактировать и сохранять этот файл нужно от имени root. Теперь остановите звуковую систему, удалите модуль, о котором идет речь, из работающего ядра, перезапустите *ALSA* и проверьте, что модуль, который только что был добавлен в черный список, отсутствует:

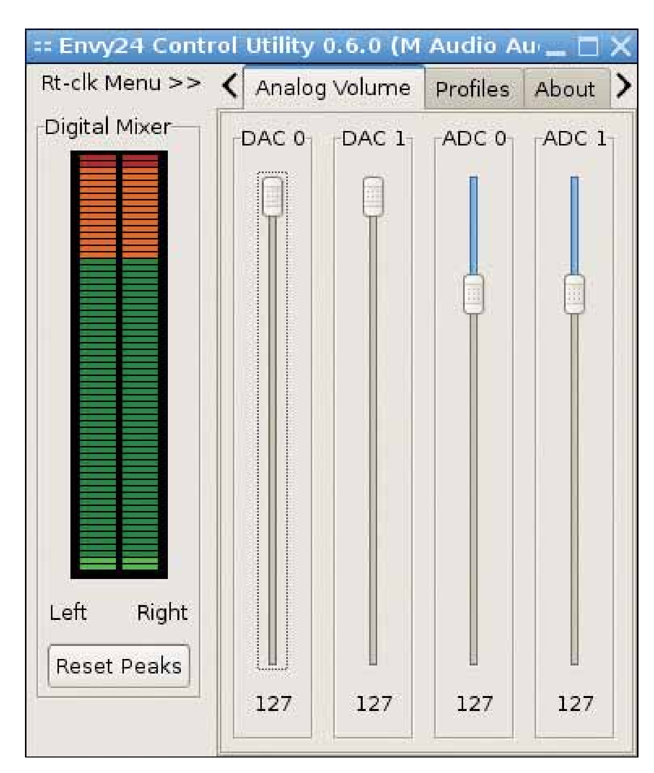
```
# /etc/init.d/alsa-utils stop
Shutting down ALSA...done.
# rmmod snd-atiixp-modem
# /etc/init.d/alsa-utils stop
Setting up ALSA...done.
# lsmodlgrep snd
```

При следующей запуске компьютера драйвер winmodem уже не будет загружаться автоматически.

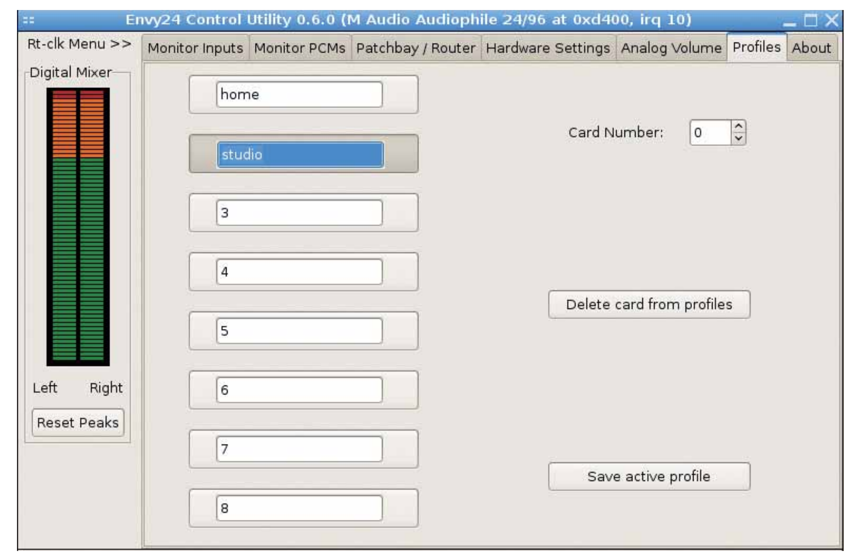
Изменение порядка звуковых карт

Теперь в системе остались только нужные нам аудиоустройства, и пора определить порядок их представления. Это поможет решить проблему унаследованных приложений с **/dev/dsp**, но удобно и для сьемных устройств. По умолчанию, при подключении устройства *ALSA* каждый раз дает ему новый индекс, и это может сбить с толку приложения, сохраняющие настройки *ALSA* не по его полному имени. Скажем, Ekiga запоминает, что для приема звонков вы используете 'Plantronics Headset', но другие программы могут хранить лишь номер: **hw:0** или **hw:1**.

Чтобы исправить это, добавьте следующие строки в конец файла **/etc/modprobe.d/alsa-base** или его эквивалента (от имени пользователя root):



Установка корректного уровня аналогового сигнала предотвратит искажения.



Профили микшера можно сохранить и потом вызывать.

```
alias snd-card-0 snd-emu10k1
options snd-emu10k1 index=0
alias snd-card-1 snd-rme9652
options snd-rme9652 index=1
alias snd-card-2 snd-usb-audio
options snd-usb-audio index=2
```

В этом примере карта Creative Labs Audigy выбрана как **card 0** для системных уведомлений, что соответствует **/dev/dsp/** в OSS. RME Hammerfall – это card 1, по этому имени звуковой сервер Jack достоверно сможет его найти; а MIDI-коннектор USB – это card 2.

Если вам все же потребуется драйвер winmodem, то есть альтернатива помещению его в черный список. Это установка значения индекса в **-2**, например:

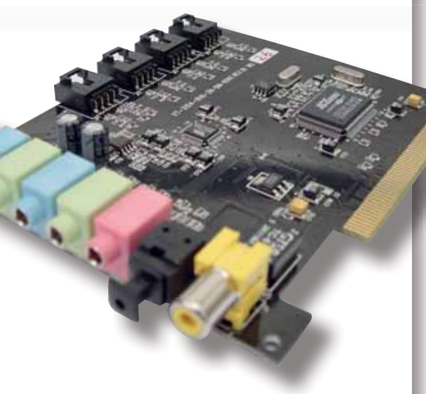
```
options snd-via82xx-modem index=-2
```

Данная опция не дает модему захватить первое устройство *ALSA* с индексом 0 и используется по умолчанию в файле **/etc/modprobe.d/alsa-base** в Debian для трех драйверов winmodem'ов, которые мы рас-

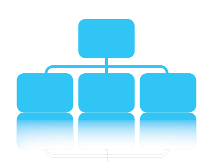
Intel HDA и ALSA

Стандарт Intel, High Definition Audio, должен заменить старый чипсет AC97, установленный на большинстве материнских плат со встроенным звуком. HDA имеет технические преимущества перед AC97, включая более высокий битрейт интерфейсов S/PDIF, множество независимых потоков, до восьми каналов и варьируемые входы/выходы (что Intel называет «переназначение разъемов», 'jack retasking'). Однако на большинстве материнских плат с HDA не реализована полная спецификация стандарта.

Так как все реализации HDA отличаются, то создание общего драйвера *ALSA*, способного хорошо работать на всех платах с HDA – задача непростая. Если с устройством HDA возникают проблемы, проект *ALSA* рекомендует перед отправкой отчета об ошибке собрать информацию из файлов **/proc/asound/card#/codec***. А если вы хотите помочь в разработке драйвера *ALSA* для HDA, воспользуйтесь утилитой на странице http://helllabs.org/codecgraph, которая может сгенерировать информацию о кодеке в формате *Graphviz*. Если вашего чипсета HDA нет на этой странице, разработчики *codecgraph* хотели бы узнать об этом. С подробностями.



Мороки со встроенной звуковой картой можно избежать, купив внешнюю аудиоплату вроде этой. Ноль проблем!



Программируем в Облаках

Хотите иметь огромный Linux-кластер и скидывать на него решение самых насущных проблем? «Большие» вычисления теперь дешевы, и это покажет **Дэн Фрост**.

Наш эксперт

Дэн Фрост – технический директор агентства web-разработок 3ev в Брайтоне. Он создает ПО для организаций, среди которых Государственная служба здравоохранения, The Economist, Virgin/EMI. В настоящий момент работает над CMS-приложением Web 2.0 в 3ev.

Время – одиннадцать утра. Вы получили письмо от директора по маркетингу: «Привет. Мы только что пообещали отправить наш дневной бюллетень 2 000 000 клиентам наших партнеров. Я сказал, что все будет ОК, и они получат его где-то через час. Когда сделаешь, сообщи. Спасибо».

Спешное деление в уме 2 000 000 жаждущих клиентов на 3 престарелых сервера не дает вам результата «все будет ОК». Конечно, можно бросить на проблему 200 серверов EC2 и посиживать, балуясь с обоями рабочего стола, пока они глодают список имен.

Для большинства из нас быстрое масштабирование своего приложения и возможность скинуть проблему на несколько десятков серверов до недавнего времени было непозволительной роскошью, пока Amazon не открыл свои вычислительные облака для широких масс.

EC2 предоставляет столько виртуальных серверов, сколько вам нужно, взимая почасовую плату. За $0,10 – да, именно десять центов США – вы получите "1 EC2 Compute Unit" [1 вычислительный модуль EC], имеющий 160 ГБ для хранения данных на 32-битной платформе с 1,7 ГБ памяти. Суммарная стоимость выходит $72 в месяц, так что это не дешевый хостинг, но это не повод им не пользоваться. Подумайте о нем иначе: за $0,40 в день вы сможете получить 4 машины, загруженные отправкой ваших дневных бюллетеней.

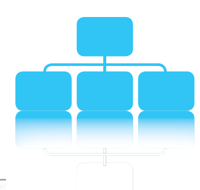
Облако? А что это?

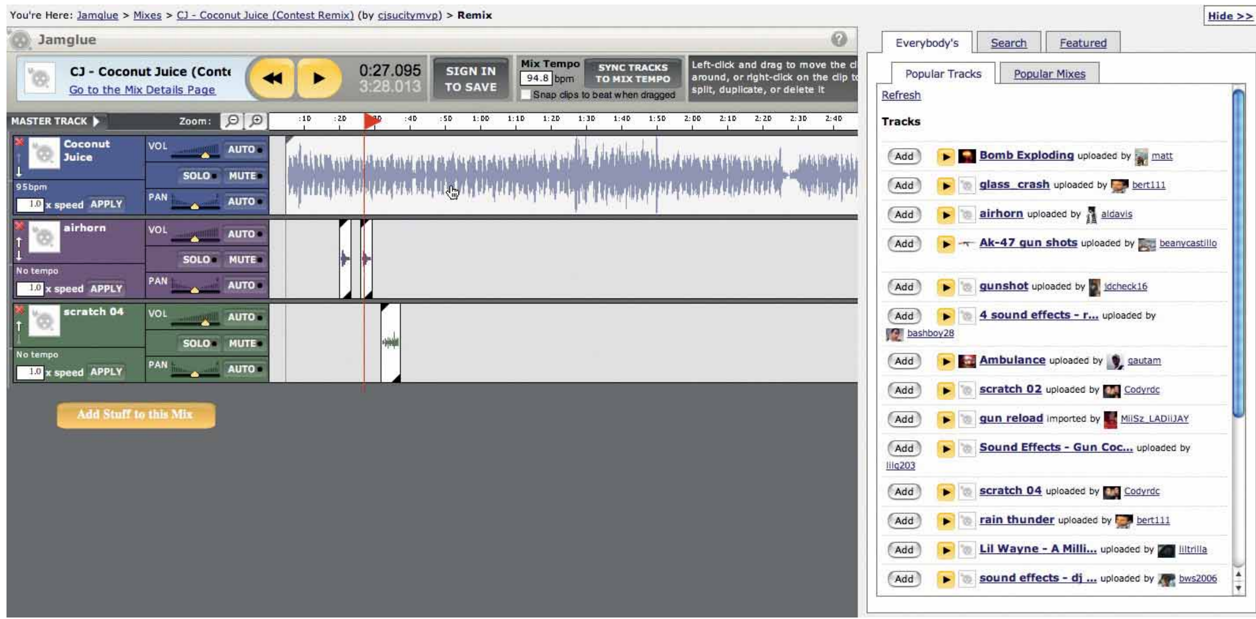
«Облаком» [cloud] прозвали огромное количество серверов Amazon, работающих в вычислительных центрах по всему миру; поверх них находится виртуализация Xen, предоставляющая тысячи виртуальных машин. EC2 существует почти два года, и уже имеет некоторые впечатляющие применения, от простого срочного масштабирования до развитой обработки данных.

New York Times применила EC2 для обработки 11 миллионов изображений статей, конвертировав их в PDF за 24 часа. Hadoop – открытая реализация алгоритма MapReduce от Google, развернутая на EC2 – позволяет выполнить колоссальный объем поисковых операций, при мизерных вложениях и трудозатратах.

Сайты вроде jamglue.com используют EC2 для внутренней обработки огромного количества аудиофайлов, на которую ушли бы длинные шеренги стоек с оборудованием. Доступ к дешевой вычислительной мощности делает построение таких систем программными, а не аппаратными проектами.

Ученые используют EC2 как дешевый инструмент для выполнения ряда «числодробильных» задач. Автоматическое тестирование придет на ум не в первую очередь, но *selenium* – инструмент как раз такого класса – может быть запущен на множестве модулей EC2, каждый из





› *Jamglue* – классное web-приложение, использующее S3 для хранения файлов и EC2 для обработки аудио.

> «...мы имеем огромную, широко доступную систему распространения контента, подключаемую к нашим “серверам по запросу” бесплатно.»
> Гаутам Джаяраман, Jamglue

которых будет работать над конкретным тестом. И если раньше прогон всех тестов для вашего приложения занимал два часа, то теперь он выполнится за 12 минут при использовании десяти «экземпляров» [instances] EC2.

Типичный пример, который мы далее рассмотрим на практике, это отправка электронной почты. Обычно, при необходимости рассылать тысячи писем ежедневно, вы отводите на эту задачу отдельные серверы, и весь остаток дня они лодырничают; а с EC2 вы просто мобилизуете 3 экземпляра EC2, скажем, в 11 утра, загружаете ваши сообщения на них и отправляете почту в полдень. Это может стоить 1–2 доллара в день – сравните с сотнями долларов, необходимых на мощный выделенный сервер со всеми сопутствующими прибамбасами.

Проще говоря, EC2 – это возможность решать более крупные задачи и масштабироваться быстрее и дешевле.

Что там за Облаками?

Итак, вот что лежит в основе, но с нашей точки зрения все выглядит гораздо проще. Подключившись к EC2, вы увидите нечто вроде обычного Linux-сервера. К нему можно обратиться по SSH, активировать сервисы, держать web-сайты, запускать процессы, точно так же, как при загрузке нормального сервера. Когда экземпляр работает, вы подключаетесь к нему либо через временный URL, либо, как было сделано недавно, через статический IP.

Если вам понадобится больше машин, вы, используя SOAP, командную строку или графический интерфейс, отправляете запрос на них, и через некоторое время получаете новые сервера для ваших нужд. Между экземплярами EC2 или между EC2 и внешним миром можно обмениваться данными.

Закончив работу с экземплярами, вам просто надо выключить их – и все, вы больше ничего не платите. Принцип «плати-пока-работаешь» означает, что вы можете тестировать сложные кластерные установки, бросать сервера на подмогу, а то и просто поиграть этак со 100 серверами.

На самом деле их тут нет...

Но не обольщайтесь: в реальности машин не существует. Экземпляры EC2 – это виртуальные серверы, и поэтому они временны. При выключении экземпляра все данные теряются; то же самое происходит при его сбое.

В чем тут смысл? Ну, сервера EC2 дешевы – считайте их еще и одноразовыми. В отличие от выделенного сервера, рядом с которым вы проводите полжизни, резервируя и занимаясь мониторингом и проверкой, экземпляру EC2 этого не нужно, потому что вы моментально можете заменить его другим.

К ним надо немного попривыкнуть, и это может изменить способ написания ваших приложений, но главное, они вселяют надежду на то, что у вас не будет точек сбоя. Серверы EC2 – просто «рабочие лошадки», а важные данные спокойно хранятся вдали от них. Если «рабочая лошадка» рухнет, данные не будут утеряны.

Как вы увидите далее, это вносит изменения в планирование и построение ваших приложений.

Начало

Во-первых, заведите себе учетную запись на Amazon Web Services: https://aws-portal.amazon.com/gp/aws/developer/registration/index.html. Затем перейдите на сайт EC2 www.amazon.com/ec2/ и с помощью кнопки “Your Web Services Account” выберите пункт AWS Access Identifiers. Вам потребуется Access Key ID и Secret Access Key с этой страницы.

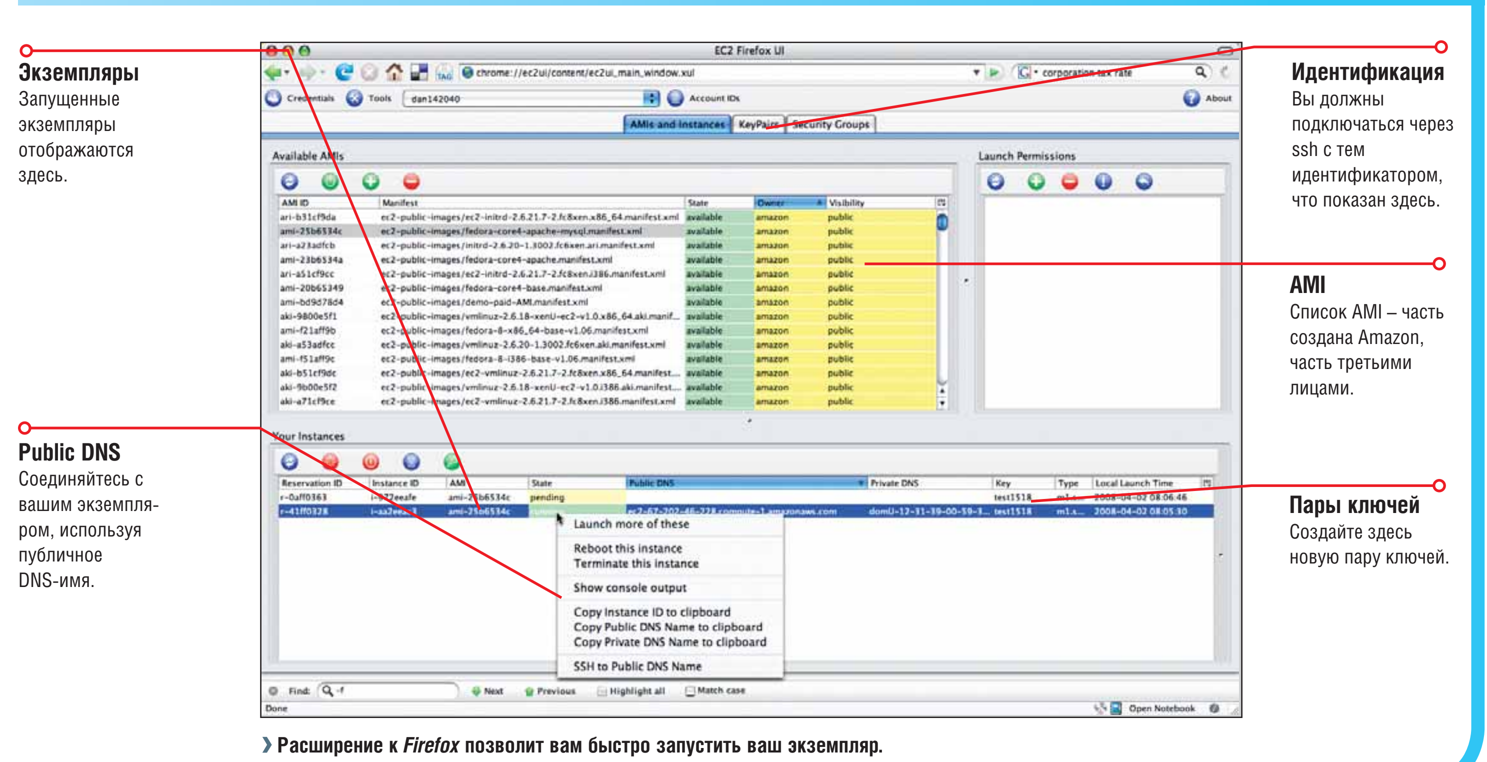
Вы можете управлять вашей частью облака EC2 через интерфейс командной строки, API SOAP или графический интерфейс. Мы же справимся с использованием EC2 при помощи расширения для *Firefox*, »

Куда это годится

Используйте это для:
- Быстро масштабируемых приложений
- Масштабирования в заданное время
- Задач массовой обработки

Не используйте это для:
- Хранения своих данных (возьмите что-то типа S3)
- Хостинга маленьких сайтов – он обходится дороже

Контроль облака через Firefox



› **Расширение к *Firefox* позволит вам быстро запустить ваш экземпляр.**

» а для любителей грязной работы есть горы документации.

Следуя инструкции с форума, используем Key ID и Secret Access Key как подтверждение ваших полномочий: http://developer.amazonwebservices.com/connect/thread.jspa?threadID=17717&tstart=45

Установив все это, вы увидите список доступных вам AMI – образов машин. Чтобы начать работать, создайте экземпляр с подходящим образом, например, F**edora-core4-apache-mysql**.

Когда образ выполняется, он возникает в списке 'Your Instances' [Ваши экземпляры]. Подождите, пока колонка Public DNS сменится с 'pending' [ожидание] на 'running' [работа].

Откройте SSH-терминал и войдите в экземпляр:

```
ssh -i ваш_ключе.pem root@публичное_доменное_имя_сервера
```

Флаг -i велит ssh использовать реквизиты ssh, созданную вами при настройке расширения *Firefox*.

Войдя в свой первый экземпляр EC2, осмотритесь вокруг. Заметим, что он по виду и ощущениям выглядит как обычный сервер, поэтому долго привыкать не придется.

› **Заполучив работающий сервер, подключитесь к нему через ssh, запустите и чувствуйте себя как дома.**

```
~/ec2$ ssh -i mykey.pem root@ec2-75-101-212-108.compute-1.amazonaws.com
Last login: Sun Jun  1 13:01:41 2008 from 86.170.74.188

         __|  __|_  )  Rev: 2
         _|  (     /
        ___|\___|___|

 Welcome to an EC2 Public Image
                      :-)

    Apache2+MySQL4

    __ c __ /etc/ec2/release-notes.txt

[root@domU-12-31-39-00-79-21 ~]# ls -l
total 0
[root@domU-12-31-39-00-79-21 ~]# /etc/init.d/httpd start
Starting httpd:
[root@domU-12-31-39-00-79-21 ~]# vi /var/www/
cgi-bin/ error/   html/    icons/
[root@domU-12-31-39-00-79-21 ~]# vi /var/www/html/index.html
```

Для примера: зайдите в **/var/www/html/** и создайте там файл **hello.php**:

```
<?php
echo "<h1>Hello, virtual world</h1>";
phpinfo():
?>
```

Скопируйте Public DNS в браузер, чтобы увидеть результат. После этого попробуйте установить что-то вроде *wordpress*, чтобы убедиться, как просто пользоваться EC2.

Теперь у вас есть свой модуль EC2; платите за него $0,10 в час и используйте как нормальный web-сервер. Если вы уже баловались с Linux-серверами, там для вас будет очень мало нового, и вы, вероятнее всего, сможете сразу начать работу. Но далее мы видим важную разницу: остановите экземпляр, например, выбрав его и нажав кнопку Terminate. Затем запустите новый экземпляр. Все, что вы делали, будет потеряно, потому что этот экземпляр абсолютно новый, а старый стерт с лица земли.

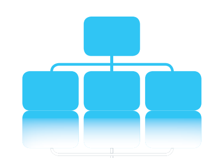
2 000 000 электронных писем

Так что же нам делать с заданием директора по маркетингу и его двумя миллионами писем? Объединим EC2, SQS и чуть-чуть Ruby on Rails.

EC2 предоставляет семейство web-сервисов для хранения данных, постановки их в очередь, поиска и прочего, обладающих широкими возможностями масштабирования, и в то же время очень простых в использовании. Один из его наиболее известных представителей – S3: многие компании применяют его для дешевого резервного копирования. Другой – Simple Query Service (SQS), и он гораздо внушительнее, чем его имя. Только представьте: 10, 100 или 1000 экземпляров EC2 все трудятся над одним центральным списком задач; простое привлечение большего числа экземпляров выполнит эту работу быстрее.

Используя очень простую модель Rails под названием 'Email', добавим несколько методов, которые будут помещать каждую запись Email в очередь SQS при сохранении:

```
# Standard Rails callback
def after_save
```



Куда это годится

Использовать чужое AMI не так-то просто, потому что экземпляры EC2 все настроены под вас.
Например, *selenium grid* использует EC2 для автоматизации тестирования web-приложений: http://selenium-grid.openqa.org/
Или используйте *Hadoop* для запуска по-настоящему крупных вещей: http://wiki.apache.org/hadoop/AmazonEC2
Попробуйте учебники и приложения на: http://developer.amazonwebservic es.com/

```
send_email_to_queue()
end
# Send the current email to the SQS queue to be delivered
def send_email_to_queue
q = Email.get_queue()
q.send_message self.to_xml
end
```

Теперь, имея возможность добавлять письма в центральную очередь, мы хотим, чтобы какой-либо из наших экземпляров EC2 брал их для отправки. Опять же, добавим это к модели e-mail:

```
# Get an email from the queue - pop from stack
def self.get_email
q = Email.get_queue()
queueitem = q.receive_message
remote_email = Email.new()
remote_email.from_xml queueitem.body
return remote_email
end
```

При создании каждого нового элемента Email он помещается прямо в SQS-очередь, где его может забрать любой экземпляр EC2:

```
e = Email.new()
e.recipient = 'dummy@example.com'
e.save()
```

И затем любой другой экземпляр в нашем коде может просто вытащить письмо из SQS и послать его:

```
e = Email.get_email
```

Нужны еще кое-какие методы, которые я здесь опустил, но и эти строки показывают, как сделать вашу задачу простой и масштабируемой. Для развертывания этого приложения нам нужен подходящий AMI – выберите из списка образ **ami-5c0aef35** (манифест ec2onrails). Получив работающий экземпляр EC2, разверните Rails-приложение на каждом сервере и запустите их.

Далее каждый экземпляр EC2 начинает забирать почтовые сообщения из очереди SQS, подготавливать их и отправлять. Rails-приложение будет просто повторять попытки, пока очередь не опустеет, так что можете добавить 10, 20, 30 экземпляров.

Иметь экземпляры EC2.0, работающие с SQS-списком задач, очень эффективно для процессов, которые могут быть децентрализованы. Пакетная отправка электронной почты – один из таких примеров, как и задача конверсии образов статей NYT. В обоих случаях большую работу можно разбить на множество более мелких, независимых задач.

А дальше что?

Для всех этих примеров нам не пришлось писать какого-либо кода с учетом окружения EC2. Пример PHP – всего лишь простое PHP, и приложение Rails не общается с EC2 каким-либо специальным образом – только через SQS. Так происходит потому, что для большинства готовых web-приложений вам не нужно менять многое для работы на EC2: машины спроектированы так, чтобы на вид не отличаться от обычных серверов.

API имеются для наиболее популярных языков – PHP, Java, Ruby, Python и т.д. – так что вы сможете сделать что-то устанавливаемое и запускаемое. Но поиграть стоит. Если ваш основной заработок – программирование, с EC2 вы сможете вбросить огромную вычислительную мощность в проблемные задачи за очень небольшие деньги. Если вы просто любите пробовать новые технологии, EC2 доставит вам большое удовольствие. Вы сможете запустить приложение такой сложности, о которой из-за убогого компьютера даже и мечтать не смели, или наконец завершить *NetHack* на суперкомпьютере! 





› **Animoto использует умные кинематические AI-технологии для создания музыкальных клипов на лету. Для рендеринга он комбинирует собственный сервер и EC2.**

А вы бы что сделали?

Возможно, Amazon'у было легче отдать EC2 в общий доступ, чем потребителям придумать хороший способ применить такое невероятное количество вычислительных ресурсов. Помимо научных потребностей, для чего еще можно его использовать? Мы рассуждаем на эту тему на первой странице, но если вы, прочитав все ответы, по-прежнему знаете то, чего не знаем мы, мы тоже хотим быть в курсе – ну для чего можно использовать EC2? Присылайте ваши ответы на letters@linuxformat.ru или www.linuxforum.ru.

Что за штука... T2



Это похоже на Gentoo, но не совсем. **Пол Хадсон** расскажет, что такое мета-дистрибутив…



На сей раз я знаю, что это такое! Это вроде Gentoo, правда?

И да, и нет. Вы, наверное, прочитали строчку под заголовком?



Ну да… И все равно мало что понял. Может, объясните подробнее?

Хорошо, начнем с Gentoo. Он и его производные (наиболее известный из них – Sabayon) – это самые популярные дистрибутивы Linux на базе исходных текстов. То есть, в Gentoo (в отличие от Ubuntu, Fedora и прочих «обычных» дистрибутивов) все программы компилируются пользователем из исходных кодов. Это совсем не так трудно, как кажется: в систему встроено множество скриптов, автоматизирующих процесс. В результате получается дистрибутив, скроенный под конкретную конфигурацию оборудования, тогда как стандартные наборы используют, в основном, универсальные пакеты для Pentium/Pro. Исключение – 64-битный режим, где Gentoo дает небольшой выигрыш (всего лишь подстраивает приложения под ваши потребности).

> «Смысл T2 в том, чтобы вы запросто собирали свой дистрибутив сами.»

Так, ясно. А при чем здесь T2?

T2, в составе «родительского» проекта Rock Linux, существовал еще до Gentoo, и хотя они оба основаны на исходных кодах, Rock Linux продвигал себя как «комплект для сборки дистрибутива». Ныне T2 использует термин «окружение для разработки системы», но идея та же: по сути, это не дистрибутив, готовый к использованию (хотя при желании и это доступно), а просто коллекция исходных текстов, из которой мощная система сборки позволяет сгенерировать дистрибутив по своему вкусу.



А не лучше ли взять что-нибудь готовое, хотя бы Debian?

Подумайте. Команда Debian уже все за вас решила: и какие темы будут у вас на рабочем столе, и какими языками оснастить Firefox. Даже сам Firefox переименовали за вас! Они решили, какие зависимости будет иметь OpenOffice.org; решили, под какой процессор оптимизировать KDE; и так далее. Поэтому, когда Debian предлагает выбрать состав пакетов для установки, вы только полируете чужую работу, а не строите дистрибутив собственными руками. А если захочется оптимизировать KDE под собственный процессор (например, новейший Intel Core), понадобится столько работы, что собрать персональный дистрибутив окажется гораздо проще.

T2 это исправит?

Конечно: у разработчиков T2 обширная коллекция программ (около 3000 наименований). Все исходники идут в проект прямо от разработчиков, поэтому ПО наисвежайшее. Команда T2 накладывает заплатки только в случае угрозы безопасности или для добавления кроссплатформенности. Но по большей части вы получаете чистое нетронутое ПО – не залапанное сотней посредников, а самую основу, где вы сами принимаете главные решения и настраиваете все «под себя».

И команда T2 обеспечит меня всем необходимым для этого?

Конечно! В этом весь смысл дистрибутива. Как я уже сказал, главные решения вы принимаете сами. На высшем уровне вы выбираете «цель» – оборудование, на котором будет работать ваш Linux. Тут свобода полная, но далее ассортимент пакетов будет соответственно ограничен. Согласитесь, KDE на Psion – это нонсенс! Выбрав целевую машину (или пропустив этот шаг, если у вас собственная цель), вы указываете архитектуру – к примеру, оставить старый добрый x86, или лучше будет x86-64? Конечно же, в выборе участвуют Alpha, ARM, IA64, 68K, MIPS, PowerPC, SPARC и иже с ними, и этот шаг снова сужает выбор пакетов. Пройдя два этих этапа, вы можете либо запустить процесс компиляции и вернуться назавтра, либо приступить к реальной работе по подбору ПО «под себя».

Представляю, насколько тяжела эта работа…

Да, нелегка, но и отважится на нее далеко не всякий. И сборочная система T2 делает все от нее зависящее, чтобы максимально упростить и облегчить процесс.

Gentoo с этим тоже неплохо справляется. Зачем же еще и T2 голову забивать?

Вспомните, что Gentoo еще и духу не было, когда появился T2? Кроме того, у T2 и Gentoo разные цели. Давайте сравним Java и Mono: Java – это виртуальная машина, где выполняются программы, и Mono как будто то же. Но виртуальная машина Java признает только Java, а Mono справляется с кодом, написанным на нескольких языках. (Заметим для пуристов: сейчас люди Java пытаются наладить поддержку других языков, но изначально этого не было). Gentoo – это все-таки дистрибутив, а не рабочая среда для его сборки. У T2 нет цели стать самостоятельным дистрибутивом: система занимает место инструктора, предоставляя человеку собственными руками конструировать для себя готовый продукт.

И дистрибутив, собранный в T2, будет похож на все остальные?

Именно так. Любую часть T2 можно заменить собственным кодом. Например, в комплекте имеется инсталлятор, но при желании вы можете взять другой. Какой-то специальной системы администрирования здесь нет, и каждый может выбрать что-то себе по душе.

Прежде чем я примусь за сборку собственной системы, не приведете ли пример успешного дистрибутива на базе T2?

Легко: Puppy Linux поначалу базировался на Slackware, а в последней версии переключился на T2. Разработчики считают, что в результате сократилось количество зависимостей, и версии пакетов теперь более свежие – а это значительное преимущество!



Круто! У меня тоже Puppy на Eee стоит — Puppeee называется...

T2 специально создан многоцелевым: и ультрамодный восьмиядерный процессор, и суперлегкий ноутбук вроде Eee – все одинаково просто, что и показывает гибкость подобных систем. В отличие от Gentoo, все в T2 исходит от единой сборочной системы и единой базы исходных кодов. В доказательство гибкости такой системы, можно собрать ее на базе ядра любой ОС – работа над Hurd, BSD, OpenSolaris и OpenDarwin завершена, на подходе поддержка Minix.

ОК, обещание поддержки Minix убедило меня окончательно. А где можно узнать об этом подробнее?

Рад, что вам понравилось! Домашняя страница проекта T2 находится по адресу www.t2-project.org. Но если вы действительно хотите узнать больше, прежде чем нырнуть в работу, советую для начала прочесть первые несколько глав превосходной онлайн-публикации T2 System Development Environment Рене Реба [Rene Rebe] и Сюзанны Клаус [Susanne Klaus], свободно доступной для чтения по адресу www.t2-project.org/handbook/html/index.html. LXF



По рецептам д-ра Брауна

Д-р Крис Браун
Доктор проводит обучение, разработки и консультации по Linux. Ученая степень по физике элементарных частиц ему в этом совсем не помогает.

Добротное администрирование систем из причудливых заворотов кишок серверной.

История повторяется?

На этой неделе, пройдя синапс между двумя из немногих уцелевших серых клеточек, в моей голове повстречались две мысли. [Синапс – область контакта нервных клеток между собой или с иннервируемыми ими тканями, – прим. пер.] Первая касалась наплыва новых релизов – Ubuntu 8.04, Fedora 9 и OpenSUSE 11. Вторая была о книге Питера Сэласа [Peter Salus] «Четверть века UNIX» [*A Quarter Century of UNIX*], где есть глава 'Duelling Unixes' («Поединок между Unix'ами»). «Война Unix» 80-х и 90-х породила производителей, конкурирующих за рынок и контроль над стандартами в этой якобы «открытой» системе. Apollo, HP, Honeywell, IBM, Intel, Microsoft, SCO, Silicon Graphics, Sun… список можно продолжить. С учетом высказываний Стива Балмера [Steve Ballmer] о Linux, трудно поверить, что в списке производителей Unix есть и Microsoft, хотя Сэлас пишет: «Не знаю, были ли инвестиции Microsoft в XENIX уловкой, чтобы >помешать< успеху Unix на рынке».

Некоторые обозреватели сочли, что именно споры между производителями позволили Microsoft со своей «новой технологией» Windows NT нащупать точку опоры на рынке серверных систем. И вот моя мысль: неужели нам снова угрожает повторение этой истории, если на те же грабли мы наступим с Linux? Может, это и не вопрос вовсе – кому, собственно, «нам»? У движения Linux нет единой стратегии на рынке, какая есть у Microsoft и какая была у Unix в девяностых.

Чего мы хотим от Linux? Чтобы он сверг с престола Microsoft и достиг мирового господства? Или остался открытой системой, поддерживающей и продвигающей инновации в плодовитом сообществе разработчиков? Решайте сами. И того, и другого сразу мы вряд ли дождемся.



Введение в Posix

Capabilities Почти удавшаяся попытка комитета Posix скорректировать старую модель Unix, где root – это «царь и бог».

Передо мной – статья Денниса Ритчи [Dennis Ritchie] (отец-основатель Unix), опубликованная в Bell System Technical Journal [внутренний журнал для научных сотрудников Bell Labs/Alcatel-Lucent, который выходит раз в квартал с 1922 г., – прим. пер.] от июля 1978 г., где он описывает данную модель – и даже эта статья называется «Взгляд в прошлое» (A Retrospective)! Модель, несомненно, довольно грубая, и Ритчи признает, что «классификация недостаточно хороша, чтобы отвечать нуждам всех инсталляций». Одно из свойств модели Unix – концепция суперпользователя root, имеющего право выполнить любую операцию и получить доступ ко всем файлам. Это очень рискованный способ предоставления привилегий, и Ритчи пишет: «Мы отдаем себе отчет в том, что сама идея суперпользователя – это теоретический, а чаще практический, недостаток любой схемы защиты». Проблема в том, что желающим сделать нечто безобидное (например, установить время) приходится давать права на все что угодно. Но как бы там ни было, эта древняя модель безопасности была унаследована Linux и отлично работает более 30 лет. Хотя было несколько попыток расширить ее.

Стандарт, которого не было

В 1997 году комитет по стандартам (POSIX 1003.1e) создал черновую версию различных расширений механизма защиты UNIX/Linux. В одном из них, названном «разделение привилегий», определялся набор из 30 специальных возможностей [capabilities], которые могут быть у процесса. Идея заключалась в том, чтобы предложить более тонкий механизм по сравнению с обычным разделением «root или не root».

Увы, комитет POSIX лишился спонсорства, и стандарт так и остался черновиком. Однако спецификация была полностью завершена, и фактически даже реализована в ядре Linux. Подробности см. в **man 7 capabilities**.

Возможности Posix навскидку

Возможность	Описание
CAP_CHOWN	Разрешить менять пользователя и группы владельца
CAP_DAC_OVERRIDE	Игнорировать проверку прав на чтение, запись и выполнение файлов. (DAC = «дискреционное управление доступом – discretionary access control».)
CAP_KILL	Игнорировать проверку прав на отправку сигналов процессам.
CAP_NET_RAW	Разрешить использование сокетов RAW и PACKET.
CAP_SYS_BOOT	Разрешить использование системного вызова reboot().
CAP_SYS_MODULE	Разрешить загрузку и выгрузку модулей ядра.

› **В таблице показаны не все возможности: просто дано представление об общей идее.**

Храните пароли в надежном месте

GPG и немного кодирования позволять вам создать надежную систему хранения паролей. Вот та, которую мы собрали для себя...

Не знаю, как у вас, а у меня столько учетных записей на разных сайтах, что приходится хранить их не только в голове. Раньше я записывал имена и пароли в маленькую черную книжечку, но всегда боялся, что она попадет в руки злоумышленника, и он серьезно испортит мне жизнь. И я решил, что будет безопаснее хранить их в зашифрованном файле: единственное, что тут надо помнить, это пароль для его декодирования. Идея такова: когда мне нужен пароль, я расшифровываю файл, беру оттуда то, что требуется и удаляю текстовую версию хранилища. Итак, мой мини-проект этого месяца – простая утилита для работы с паролями.

Для шифрования я взял *gpg*, утилиту, использующую схему с открытым/закрытым ключом (для моих целей это даже слишком хорошо), в основе которой лежит достаточно сложная технология. Но вам нужно знать лишь то, что для *gpg* надо сгенерировать открытый и закрытый ключи и сохранить их в каталоге **~/.gnupg**. С помощью открытого ключа файл с паролями шифруется, а закрытого – расшифровывается. Закрытый ключ и сам зашифрован и защищен «парольной фразой». Ее я и буду помнить, чтобы иметь доступ ко всем паролям.

Простоты ради я свернул необходимые команды в небольшой скрипт (который есть на DVD). Он называется "pwm" (управление паролями – password management) и запускается тремя способами:

```
pwm -i
```

Это инициализация системы. Для генерации открытого/закрытого ключа вызывается **gpg --gen-key**, создается начальное (естественно, пока пустое) зашифрованное хранилище паролей. Такой вызов нужен только один раз.

```
pwm -e
```

Хранилище паролей расшифровывается в текстовый файл и открывается в редакторе (*vi*). *gpg* запросит парольную фразу. При сохранении файла он сразу шифруется, а текстовая версия удаляется. Обратите внимание, что у файла нет определенного синтаксиса – это просто набор строк, которые можно пропускать через grep. Например, мою учетную запись в *Linux Format* можно записать так:

```
Linux Format caligula / L1ttleB00ts
```

Третий вариант вызова скрипта таков:

```
pwm запрос
```

Здесь я ищу пароль. Хранилище расшифровывается, и результат передается в grep для поиска заданной строки. *gpg* опять же запросит парольную фразу.



› *Revelation* **хорошо интегрируется с GNOME. (Нет, p1gstr0tters – не мой настоящий пароль.)**

Скрипт достаточно прост в работе, если не считать длинного и страшноватого диалога при начальной генерации пары ключей командой **gpg --gen-key** (его я воспроизводить не буду). В большинстве случаев можно просто использовать варианты по умолчанию. Когда будет запрошено «настоящее имя» ("real name"), нужно ввести имя пользователя (или по крайней строку, где оно есть). Это имя будет использоваться для выбора открытого ключа. Поля «адрес электронной почты» ("email address") и «комментарий» ("comment") можно оставить пустыми.

А если нужно графическое (и более изысканное) средство управления паролями, попробуйте *Revelation* (**http://oss.codepoet.no/revelation**). В *Revelation* можно работать с одним или несколькими хранилищами: в каждом могут содержаться данные о множестве сайтов и паролей, и их можно организовать в папки.

Все хранилища зашифрованы и защищены паролем, который нужно ввести для открытия. Требуемый сайт можно найти с помощью панели слева, в стиле проводника, или поля поиска. Как видно на экранном снимке, пароль будет явно показан в окне *Revelation*, хотя эту возможность можно отключить. После выбора ресурса, нажатие на кнопку Перейти на сайт (Go to) отправит вас на него; соответствующий пароль (в чем и фишка) будет скопирован в буфер обмена, готовый ко вставке.

Revelation – приятное и хорошо продуманное приложение. Меня беспокоит только одно: если оно будет запущено постоянно (а так удобнее всего), то любой подошедший к вашему компьютеру, пока вы пьете кофе в сторонке, увидит имена и пароли. Это почти то же самое, что оставить мою маленькую черную книжечку на столе открытой.

Взгляд в виртуальность

Перейду от глобальных вопросов к мелочам: недавно мое внимание привлекла еще одна интересная штука – JeOS (вероятно, произносится как 'juice'). Это «обкромсанный» Linux-сервер от Canonical, предназначенный для построения «виртуальных приставок» [appliances] – полных образов компьютеров для запуска в виртуальной среде типа VMware. Для JeOS достаточно 128 МБ памяти и менее 300 МБ жесткого диска. Это серверная система (без графической среды) с ядром 2.6.24, специально оптимизированным для VMware. Если вам интересно взглянуть на нее, загрузите ISO-образ (менее 100 МБ) с сайта **http://cdimage.ubuntu.com/jeos**.



AppArmor

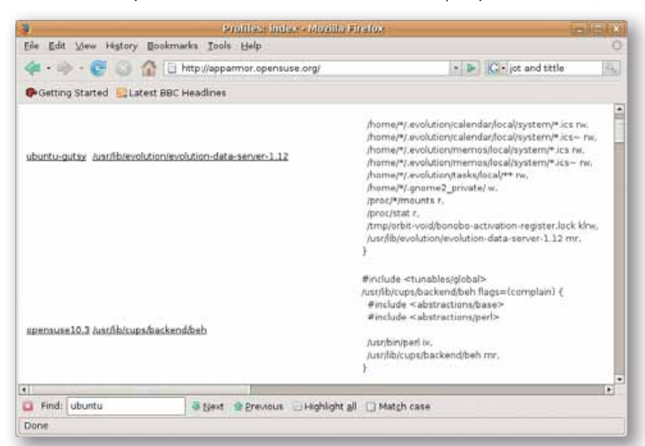
Посадите свои приложения на карантин – для их же блага и чтобы защитить систему.

Один из проектов, основанных на черновиках Posix – это *AppArmor*. Изначально разработанный Immunix и купленный Novell в 2005 г., он был включен в SUSE Linux, а также поддерживается в Ubuntu и Mandriva (Red Hat вместо этого выбрал SELinux). По сути, *AppArmor* – это профиль на базе ядра, закрывающий программе доступ к не относящимся к ней ресурсам и указывающий, какие возможности POSIX ей требуются для нормальной работы. Профили хранятся в текстовых файлах в каталоге **/etc/apparmor.d**. Имя профиля совпадает с именем исполняемого файла, к которому он относится, но слэши заменены точками. Например, профиль для **/usr/sbin/traceroute** (сетевой утилиты, выводящей путь, по которому проходит пакет, чтобы достичь заданного адреса в Интернете) находится в файле **/etc/apparmor.d/usr.sbin.traceroute**. Давайте заглянем в него:

```
1. #include <tunables/global>
2. /usr/sbin/traceroute flags=(complain) {
3. #include <abstractions/base>
4. #include <abstractions/consoles>
5. #include <abstractions/nameservice>
6. capability net_raw,
7. /usr/sbin/traceroute rmix,
8. @{PROC}/net/route r,
10. }
```

Строка 2 говорит, что данный профиль принадлежит программе **/usr/sbin/traceroute**, и что при загрузке его в ядро он будет работать в неактивном режиме complain [англ. «жалоба»]. Другими словами, *AppArmor* не будет принуждать к выполнению политики, но занесет в файл журнала все ее нарушения. Это также называется режимом обучения – почему, узнаем чуть позже. В строках 3, 4 и 5 подключаются правила из трех других файлов в каталоге abstractions. Будь мы «в объектно-ориентированном настроении», мы сказали бы, что профиль traceroute наследуется от этих трех базовых классов. *AppArmor* охотно подключает файлы, чтобы «вынести за скобку» общие наборы ресурсов; например, в файле **abstractions/nameservice** содержатся правила для разрешения DNS, LDAP, NIS, SMB, поиска реквизитов пользователей и групп в парольных файлах, которые являются общими для многих программ.

Самое сердце профиля traceroute – строки 6 и 7. Шестая строка говорит, что программа может использовать возможность net_raw. Это одна из тех возможностей POSIX, о которых мы говорили выше; там она называется CAP_NET_RAW. Данная возможность разрешает программе использовать **raw**-сокеты, то есть собирать собственные IP-датаграммы. (Ей это необходимо: при работе traceroute корректирует поле TTL – «время жизни» – заголовка IP-пакета.) Строка 7 сообщает, что программе нужен доступ к файлу **/usr/bin/traceroute** – **rmix** определяет требуемое сочетание режимов доступа и по существу выдает программе права на чтение и выполнение этого файла. Строка 8 говорит, что программе нужен доступ на чтение к **/proc/net/route**. (Обозначение **@{PROC}** ссылается на переменную, определенную в **<tunables/global>** и задающую точку монтирования файловой системы **procfs**. В любом Linux это почти наверняка **/proc**.)



› «Биржа» *AppArmor* **предлагает около 300 готовых профилей.**

Другие дистрибутивы

Я запускал свой пример на Ubuntu 8.04, но профили и команды *AppArmor* должны выглядеть почти так же в любом другом дистрибутиве. В SUSE добавлен ряд графических оболочек вокруг утилит *AppArmor*, работающих из командной строки; они доступны через *YaST*. На соседней странице показан экранный снимок *YaST* из OpenSUSE 10.3, идентичный описанному здесь диалогу *Genprof*.

В каждую мелочь можно не вникать, но общая идея понятна: профиль определяет, к каким ресурсам программа должна иметь доступ. Подчеркнем, что профили *AppArmor* не заменяют и не обходят основной механизм управления правами в Linux. Например, включение net_raw в профиль traceroute не предоставит программе данной привилегии, если ей уже отказано в этом. Но ИСКЛЮЧЕНИЕ возможности net_raw из профиля ОТКЛЮЧИТ эту возможность, даже если она имелась ранее (например, программа была запущена пользователем root).

Для активации профиля нужно принудительно загрузить его в ядро. Сделайте это с помощью команды enforce:

```
$ enforce traceroute
Setting /etc/apparmor.d/usr.sbin.traceroute to enforce mode.
```

Обычно это происходит во время старта в init-скриптах. Обратно в режим complain профиль переключается командой complain:

```
$ complain traceroute
Setting /etc/apparmor.d/usr.sbin.traceroute to complain mode.
```

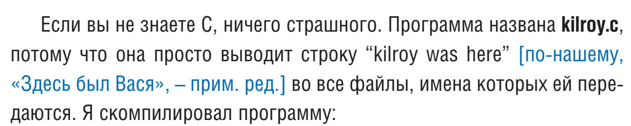
Чтобы увидеть, какие профили сейчас загружены, выполните скрипт запуска **apparmor** с аргументом “**status**”. Вот несколько строк из этого примера:

```
$ sudo /etc/init.d/apparmor status
apparmor module is loaded.
14 profiles are loaded.
4 profiles are in enforce mode.
/home/chris/kilroy
/bin/date
/usr/sbin/traceroute
/usr/sbin/avahi-daemon
10 profiles are in complain mode.
/usr/sbin/ntpd
/usr/sbin/identd
/usr/sbin/nmbd
... СТРОКИ ОПУЩЕНЫ ...
```

Откуда берутся профили?

Да, откуда же они берутся? Ну, можно усесться за исходный код программы, обмотав голову мокрым полотенцем, определить, какие ресурсы требуются и создать профиль «вручную»: ведь это всего лишь текстовый файл. А можно поискать готовый вариант среди включенных в ваш дистрибутив Linux. Например, в Ubuntu в пакете *AppArmor*-profiles их около ста. Имеется также «биржа» профилей (сайт **http://apparmor.opensuse.org**). Но всего интереснее, что *AppArmor* умеет создавать профили сам, если приложения запускаются в режиме «обучения». Чтобы показать, как это работает, создадим профиль для несложной программы на C:

```
#include <stdio.h>
int main(int argc, char *argv[])
{
 int i;
 FILE *fd;
 for (i=1; i<argc; i++) {
  fd = fopen(argv[i], “w”);
  if (fd == NULL) {
   fprintf(stderr, “failed to open %s\n”, argv[i]);
   return 1;
  }
  fprintf(fd, “kilroy was here”); fclose(fd);
 }
}
```

Если вы не знаете C, ничего страшного. Программа названа **kilroy.c**, потому что она просто выводит строку "kilroy was here" [по-нашему, «Здесь был Вася», – прим. ред.] во все файлы, имена которых ей передаются. Я скомпилировал программу:

```
$ cc kilroy.c -o kilroy
```

и если запустить ее так:

```
$./kilroy foo bar
```

то она откроет файлы **foo** и **bar** и запишет в них строку. В терминах *AppArmor*, сейчас программа работает «свободно», так как для нее нет профиля, загруженного в ядро. Она подчиняется только традиционному контролю доступа к файлам в Linux.

Чтобы сгенерировать профиль для *kilroy*, выполните команду **genprof** (как root) с именем программы в качестве аргумента:

```
# genprof /home/chris/kilroy
```

Genprof загрузит профиль в ядро в режиме complain и предложит «Запустить программу в новом окне и проявить ее функционал» ("Start the application to be profiled in another window and exercise its functionality now"). Идея состоит в том, чтобы сейчас выполнить полный набор обычных действий программы. Для своего игрушечного примера я просто наберу команду:

```
$ ./kilroy foo bar.txt
```

Так как *kilroy* запущена в режиме complain, то все действия, запрещенные программе, заносятся в журнал. (Поскольку профиля пока нет, заносятся вообще все действия.) Итак, в журнале сейчас три записи: маркер и открытие на запись файлов **foo** и **bar.txt**. Когда я решу, что «проявление функционала» завершено, я вернусь в окно, где запущена *genprof*. *genprof* пройдется по записям в журнале (для этого вызывается вспомогательная программа *logprof*) и спросит меня, разрешить или запретить каждую из них, а также предоставит возможность обобщить записи с помощью шаблонов. Весь диалог длинный, поэтому привожу фрагмент:

```
# genprof /home/chris/kilroy
Writing updated profile for /home/chris/kilroy.
Setting /home/chris/kilroy to complain mode.
```

Пожалуйста, запустите приложение, профиль которого нужно создать, в другом окне и выполните все обычные действия. Потом нажмите на кнопку Сканировать (Scan), чтобы найти события *AppArmor* в системных журналах. Для каждого из них можно будет разрешить или запретить доступ.

```
Profiling: /home/chris/kilroy
[(S)can system log for SubDomain events] / (F)inish
Reading log entries from /var/log/messages.
Updating AppArmor profiles in /etc/apparmor.d.
Enforce-mode changes:
.. СТРОКИ ОПУЩЕНЫ ...
Profile: /home/chris/kilroy
Path: /home/chris/bar.txt
Mode: w
Severity: 6
1 - /home/chris/bar.txt
2 - /home/*/bar.txt
[3 - /home/chris/*.txt]
(A)llow / [(D)eny] / (G)lob / Glob w/(E)xt / (N)ew / Abo(r)t /
(F)inish
Adding /home/chris/*.txt w to profile.
... МНОГО СТРОК ОПУЩЕНО ...
```

Во второй половине листинга *Genprof* сообщает, что *kilroy* открыл файл **/home/chris/bar.txt** на запись, и я решил обобщить это правило, позволив ему записывать данные в любой файл с именем **/home/chris/*.txt**. В данном случае мое решение было произвольным; но на практике нужно хорошенько подумать!

По завершении работы *Genprof* новый профиль сохраняется и автоматически переходит в режим enforce. Результат наших трудов (**/etc/apparmor.d/home.chris.kilroy**) выглядит следующим образом:

```
# Last Modified: Mon Apr 7 16:17:04 2008
```

YaST2@suse103

AppArmor Profiling Wizard
This wizard presents entries generated by the AppArmor access control module. You can generate highly optimized and robust security profiles by using the suggestions made by AppArmor. AppArmor suggests that you allow or deny access to specific resources or define execute permission for entries. These

AppArmor Profile Wizard

Profile /home/chris/kilroy
Path /home/chris/bar.txt
Mode w
Severity 6

○ /home/chris/bar.txt
○ /home/*/bar.txt
◉ /home/chris/*.txt

› В SUSE Linux есть графический мастер для управления профилями *AppArmor*, как часть *YaST*.

```
#include <tunables/global>
/home/chris/kilroy {
#include <abstractions/base>
/home/chris/*.txt w,
/home/chris/foo w,
/home/chris/kilroy mr,
}
```

Теперь программа *kilroy* ограничена своим профилем и может делать не все. Например,

```
$ ./kilroy foo apple.txt banana.txt
```

работает отлично, а

```
$ ./kilroy orange.txt cabbage
failed to open cabbage
```

не пройдет, так как открытие файла **cabbage** на запись не разрешено профилем, и вызов fopen() завершится неудачно.

Это игрушечный пример. Чтобы создать профиль для настоящей программы, придется продумать, какие ситуации прокрутить, чтобы задействовать все ее возможности. Обычно сперва с помощью *genprof* создается начальный профиль, затем он на несколько дней переключается в режим complain. Потом запускается *logprof*, чтобы отследить все нарушения и решить, стоит ли добавлять их в профиль. Наконец, профиль переключается в режим enforce.

В чем же тут польза? Допустим, программу *kilroy* каким-то образом взломали – например, атакой с переполнением буфера – и вынуждают ее сделать нечто такое, для чего она не предназначалась. Но она не сможет выйти за пределы своего профиля! *AppArmor* лучше всего работает с теми программами, чей набор операций четко ограничен. Например, почтовый сервер оперирует строго определенными файлами и каталогами, и с помощью профиля *AppArmor* его можно эффективно контролировать. Особенно полезен *AppArmor* в ограничении тех программ, которые являются «посредниками в передаче прав», т.е. приложений, которых вы просите сделать что-то от своего имени, но с правами другого пользователя. В эту категорию входит большинство сетевых серверов. Для более подробной информации прочтите man для **apparmor**, **apparmor.d** (описывает синтаксис профилей), **genprof**, **logprof**, **enforce** и **complain**, или зайдите на сайт http://en.opensuse.org/Apparmor. LXF



Карантин компьютера

В исходной документации *AppArmor* говорилось об «иммунизации» приложений, но для предотвращения взлома ничего не делается; скорее, ограничивается возможный ущерб на случай такой беды. «Карантин», пожалуй, более уместная медицинская метафора. Представьте, что *AppArmor* дает программе большой белый носовой платок – чихать в него, чтоб не сеять вокруг себя микробы.



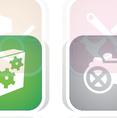




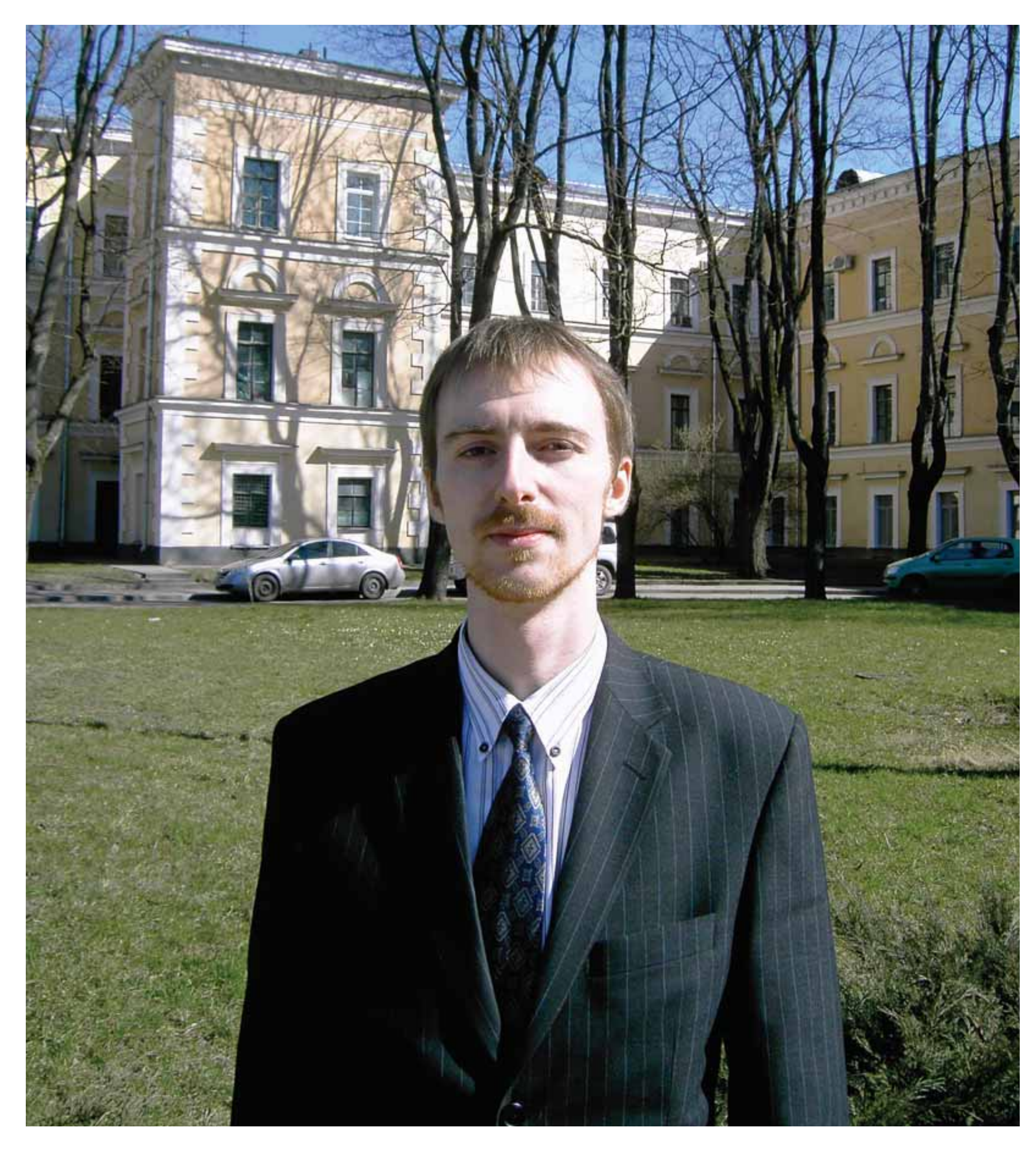








» Рубрику ведет Евгений Балдин



» **Кто**
Юрий Владимирович Колесников, начальник управления информационных технологий Администрации Псковской области.
» **Откуда**
Псковская область, областная Администрация.
» **Задача**
Поддержка компьютерного парка Администрации.
» **Результат**
Там, где нужно – автоматизируем, где необходимо – обновляем инфраструктуру, а где можно – внедряем открытое ПО.

Администрация Псковской области и ее компьютеры

Вопреки распространенному мнению, чиновники занимаются отнюдь не только усложнением жизни окружающих. По сути, они – признак цивилизованных отношений в обществе. А если общество цивилизованное, то и открытому ПО есть где развернуться – именно этим процессом и руководит **Юрий Колесников**.

Свой самый первый опыт работы с Linux я получил в далеком уже 2001 году: помогли любопытство и необходимость собрать на старом оборудовании надежный корпоративный шлюз.

В Администрации Псковской области работаю с июня 2006 года. Сказать, что компьютерный парк здесь разнообразен – это значит ничего не сказать. Число рабочих станций переваливает за 700. Они разбросаны по органам государственной власти субъекта (исполнительным, законодательным и иным), есть еще территориальные подразделения по районам области, но там обслуживание происходит по требованию и не всегда нашими силами. К счастью, основная часть компьютеров (около 500) находится в едином комплексе зданий, а остальные – в пределах областного центра. Связь обеспечена арендованными оптическими каналами. Квалификация сотрудников самая обычная. Для большинства компьютер – это пишущая машинка, но мы стараемся изменить ситуацию…

Полностью на Linux переведено пока всего 7–8 % рабочих станций, не считая ноутбуков депутатов Областного Собрания (42 машины с Ubuntu), используемых для работы с проектами документов между и

во время сессий, а также для организации голосования. Кстати, для записи звуковых стенограмм сессий Собрания мы также применяем открытое ПО: Audacity (LXF106) и Linux. Сейчас ищем решение для аннотации полученных звуковых файлов – т.е. разметки, когда и что происходило, чтобы легче было осуществлять навигацию. Если кто-то может подсказать вариант, будем премного благодарны.

На всех рабочих станциях установлены *Mozilla Firefox*, *Thunderbird*, *OpenOffice.org* и *TightVNC*. Для коллективной работы в нашем управлении используется *eGroupWare*; в данный момент начато ее внедрение и в управлении финансово-хозяйственного обеспечения. Подобная система очень полезна везде, где есть потребность в четкой фиксации заявок на обслуживание. В рамках нашего управления также эксплуатируется VoIP на базе *Asterisk* (в варианте FreePBX).

Излишне говорить, что наши серверы также управляются свободным ПО. Из редких продуктов можно упомянуть *Funambol* (**www.funambol.com/opensource**), который синхронизирует календари и контакты по протоколу SyncML, и программу для автоматической инвентаризации рабочих станций *OCS Inventory* (**www.ocsinventory-ng.org**). Потихоньку приучаем пользователей к централизованному хранению документов. Это надежнее и позволяет (в перспективе) построить систему поиска.

Кое-где применяется виртуализация на базе *VMWare* (*Xen*, по ряду причин, не прижился). Основное – это возможность резервного копирования виртуальной машины и ее быстрый перенос между серверами для распределения нагрузки или при плановой/аварийной остановке оборудования.

В порядке эксперимента мы запустили на Linux два информационных киоска: в Псковском Дворце бракосочетаний (он показывает дни и часы, доступные для подачи заявлений) и в Областной научной библиотеке (новости, поиск по каталогу и т.п.). Сама информационная сеть библиотеки частично построена на основе LTSP. Внедрение проходило силами сотрудников учреждения, мы их только консультировали. Такой подход позволил сэкономить значительные средства и оснастить библиотеку более масштабно. Первой ласточкой стало открытие виртуального читального зала Электронной библиотеки диссертаций Российской государственной библиотеки на базе 10 LTSP-терминалов.

Сегодня для полноценных рабочих станций используется Ubuntu/Kubuntu. Для тонких клиентов в качестве базового выбран SLES, но, видимо, Ubuntu скоро заменит его и здесь – по нашему субъективному мнению, это очень дружественный дистрибутив. Вся поддержка выполняется своими силами, и формальные причины этого достаточно просты: нет локальных компаний, специализирующихся на таком ПО, а иногородние для нас – не вариант. Хотя, по правде говоря, не факт, что мы сейчас решились бы отдать эту работу на аутсорсинг.

Проблем с переводом рабочих станций было немало. Основная техническая трудность – это, очевидно, совместимость с оборудованием. Нам приходилось встречать материнские платы, которые очень плохо (или совсем никак) уживались с Linux. Принтеры, сканеры и т.п. тоже приходится выбирать очень осмотрительно. В настоящий момент мы совместно с партнерами (поставщиками техники) проводим широкомасштабное тестирование и формируем типовые рекомендации для всех бюджетных учреждений области.

К сожалению, необходимо констатировать, что помимо чисто технических проблем имеется достаточно сильное сопротивление со стороны пользователей, которое преодолевается, в основном, административными методами. Основная причина недовольства – необходимость прилагать усилия (пусть и незначительные) для освоения новой среды. Иногда бывают и более принципиальные возражения, связанные с неидеальной совместимостью *OOo* и *MSO*. Скажем, режим комментирования текста в *OOo* реализован очень ненаглядно. В версии 3.0, если верить анонсам, это будет исправлено, и тогда часть проблем отпадет сама собой. Обучение пользователей происходит индивидуально, обычно прямо в процессе работы, при возникновении сложностей. Польза от потокового обучения, с нашей точки зрения, сомнительна, поскольку уровень навыков и способности к адаптации сильно различаются: кто-то привыкает сразу и без посторонней помощи, а с кем-то нужно «нянчиться» довольно долго.



В ситуации, когда необходимое программное обеспечение несовместимо с *OOo*, мы ставим *Microsoft Office*. В планах – изучить *UniOffice@Etersoft*, чтобы хотя бы частично уйти от подобных проблем. Мы достаточно плотно сотрудничаем с компанией Etersoft в плане тестирования их решений на применимость к тем продуктам, которые мы не можем заменить.

Основной глобальный выигрыш от использования открытого ПО, искупающий все связанные с этим неудобства – это значительное уменьшение количества обращений в службу поддержки, связанных с неработоспособностью системы, например, из-за вирусной активности или по вине пользователя. Это выгодно не только нам, но прежде всего самим сотрудникам Администрации. Что касается серверных решений, то из-за широкого ассортимента альтернатив и низкой стоимости ошибки, выбор в пользу открытого ПО сложно переоценить. Ведь переход с одного открытого продукта на другой (если это объективно необходимо) получается очень дешевым.

Несмотря на то, что число жалоб пользователей уменьшилось, мы отнюдь не бездельничаем. В ближайшем будущем хотим утвердить формат OpenDocument в качестве обязательного в органах власти субъекта Федерации, начали пилотный проект по использованию системы Naumen DMS (**www.naumen.ru**), готовимся к полномасштабному внедрению OpenLDAP, вплоть до авторизации входа на Windows-рабочих станциях.

Очень сильно не хватает официальной поддержки российской криптографии в OpenSSL (ожидается только в версии 0.9.9, а сроки выхода неизвестны) и сертификации этого продукта ФСБ. Это, конечно, фантастика, но вдруг получится… Запланирован также перевод на Linux максимально возможного числа рабочих станций. Это не очень стремительный процесс, но постепенно мы достигнем цели.

Естественно, как почти всем и почти везде, нам чрезвычайно не хватает денег. А если серьезнее, то основная структурная проблема связана с отсутствием единой политики в области применения ИТ в сфере государственного управления. Каждое министерство решает эти вопросы самостоятельно, а крайними оказываются субъекты, которым спускают сверху готовые решения, основанные то на *MS SQL*, то на *Oracle*, и т.д. Приходится «разводить и кормить» этот зоопарк. Умиляет также закрепление использования закрытых форматов (чаще всего – .doc, .xls) в нормативных актах различного уровня. Это очень плохая практика.

В заключение хотелось бы пожелать более активной рекламной политики стороны сообщества. Чрезвычайно полезен процесс налаживания контактов с крупными интеграторами и с государственными структурами, принимающими глобальные решения. Особое внимание хорошо бы уделить возможностям применения открытых решений в малом бизнесе – там пока еще совсем «непаханое поле». LXF

OSS в России:

В последние несколько лет вопрос о юридической стороне использования свободного ПО в России получил первостепенное значение. **Владимир Житомирский** развеет популярные мифы и даст ряд практических рекомендаций.

Исключительное право на результат интеллектуальной деятельности, созданный творческим трудом, первоначально возникает у его автора.

Правообладатель может по своему усмотрению разрешать или запрещать другим лицам использование результата интеллектуальной деятельности.

Правообладатель может распорядиться принадлежащим ему исключительным правом на результат интеллектуальной деятельности любым не противоречащим закону и существу такого исключительного права способом, в том числе путем его отчуждения по договору другому лицу (договор об отчуждении исключительного права) или предоставления другому лицу права использования соответствующих результатов интеллектуальной деятельности в установленных договором пределах (лицензионный договор).

«Винегрет» из ст.ст. 1228, 1229 и 1233 пресловутой части четвертой Гражданского кодекса РФ



Наш эксперт

Владимир Житомирский
начальник отдела особых операций и Председатель Совета директоров Правового бизнес-бюро «Граф Маевский» (www.gm-legal.com), г. Санкт-Петербург. Все компьютеры Бюро работают под управлением Linux.

В каком-то журнале в 70-е годы прошлого века была рубрика «Произведения, состоящие из одних названий». В ней, в иронической, правда, манере публиковались названия, после которых уже не надо было читать остальной текст. Вспомнил я об этом потому, что цитата из кодекса, взятая эпиграфом к данной статье, в идеале делает ненужным дальнейшие рассуждения о законности использования свободного программного продукта под лицензией GNU в России. Но мы живем не в идеальном мире, и право применяют люди. Я постараюсь избежать обильных ссылок на законы, и буду давать их только тогда, когда отсылка к ним может сыграть роль в сложной ситуации. Также примем за аксиому, что любые правовые конструкции работают при соблюдении определенных, а в нашем случае – очень простых условий.

Начнем с того, что закон (здесь и далее – российский, если не оговорено другое) устанавливает, что исключительное право на результат интеллектуальной деятельности, в том числе и при создании программ для ЭВМ, принадлежит изначально автору; автор программы в основном приравнен в правах к автору любого произведения. Гражданство правообладателя значения не имеет. И автор, как правообладатель, может по своему усмотрению разрешать или запрещать другим лицам использование результата интеллектуальной деятельности, в том числе и путем предоставления другому лицу права использования соответствующих результатов интеллектуальной деятельности в установленных договором пределах (лицензионный договор).

Встает закономерный вопрос – а где же сам договор? И это предусмотрено законом. Вот выдержка из статьи 1286 части четвертой Гражданского кодекса: *«Заключение лицензионных договоров о предоставлении права использования программы для ЭВМ ... допускается путем заключения каждым пользователем с соответствующим правообладателем договора присоединения, условия которого изложены на приобретаемом экземпляре ... программы ... либо на упаковке этого экземпляра. Начало использования ... программы ... пользователем, как оно определяется этими условиями, означает его согласие на заключение договора»*. Поскольку правообладатель не лишает себя права выдачи лицензий другим лицам, это будет простая (неисключительная) лицензия. Надо ли напоминать пользователям, что в составе практически любого свободного продукта есть текст лицензии (то есть договора присоединения) или, по крайней мере, ссылка на лицензию GNU или иную из того же ряда? А уж коробочные версии без той или иной лицензии, причем отсылающей к GNU GPL, мне вообще не встречались.

Следует подчеркнуть, что для удостоверения правомочности обладания программой закон не требует никаких наклеек, голограмм и прочего.

GPL в России

Перед тем, как коснуться GNU GPL, хочется высказать несколько соображений. Первое: мало кто из тех, кто ссылается на GPL, читал эту лицензию; большинство знает и применяет лишь поверхностно усвоенные и устраивающие их положения. Второе: я категорический противник нарушения авторских прав. Я глубоко убежден, что свободное программное обеспечение двигает человечество вперед, но я также убежден, что нигилистическое отношение к чужим правам и само наше движение погубит, и приведет к регрессу в сфере разработки ПО. Свобода и халява – суть разные вещи. Так, например, запрет копировать и распространять журнал Linux Format не есть посягательство на чьи-то мифические права, а законные действия правообладателя. Полностью соответствующие, кстати, всему букету свободных лицензий.

А теперь обратимся к лицензии GNU GPL и посмотрим, как она (любая версия) соотносится с российским законом. И убедимся, что установленные ею правила распоряжения правами отвечают отечественным. На что следует обратить внимание – в соответствии с GNU GPL, она действует, если приобретателю (в числе прочих условий) вместе с экземпляром программы передается копия именно самой лицензии GNU. Что, заметим, опять же соответствует требованиям российских законов. Попросту говоря, сам текст лицензии GNU, тем или иным образом связанный с программой (или, на худой конец, хотя бы ссылка на него), делает продукт легитимным. А основные прочие условия – на каждом экземпляре помещен знак охраны авторского права и уведомление об отсутствии гарантий; оставлены

Взгляд правоведа

без изменений все уведомления, относящиеся к самой лицензии и отсутствию гарантий.

Как-то в одной дискуссии я услышал такой аргумент: продукты Windows в России зарегистрированы, а свободные программы нет. Ответ простой: закон дает правообладателю **право** (простите за невольную тавтологию) **по собственному желанию** зарегистрировать программу для ЭВМ (ст. 1262 ГК). Такая регистрация, кстати, осложняет переход всяких прав, их в этом случае также нужно регистрировать. Так что не думаю, что Microsoft регистрирует свои продукты.

Итак, вывод – лицензия GNU GPL российским законам не противоречит и, следовательно, ее действие в России законно.

От слов – к делу

Это была теория, а она, как известно, без практики суха. Перейдем к жизни. Сначала поговорим о проблеме, кажущейся многим серьезной – о проблеме, которой на самом деле не существует. Периодически слышу сетования, что, мол, лицензии GNU в России официально не признаны. Господа, расскажите мне, пожалуйста, что такое официальное признание лицензии (по своей сути, договора), в чем оно выражается, лицензии каких правообладателей официально признаны и, опять же, в чем это выразилось? Боюсь, что ответа не дождусь.

Любой документ в любой стране признается (слово «официально» в данном контексте совсем ни при чем), если он надлежащим образом оформлен и предъявлен (да, педанты могут сделать тут некоторые замечания – не относящиеся, однако, к исследуемой ситуации). В России официальным языком общения является русский, и понятно, что все пускаемые в оборот документы должны быть на русском языке. И где здесь проблема? Любой юридически значимый документ переводится у нотариуса переводчиком, чьи знания проверил этот нотариус, после чего он, нотариус, ставит на перевод свою красивую печать. И все. Сомневается нотариус – сделайте снимок экрана сайта с текстом нужной вам лицензии. Наши клиенты, да и я сам, имели дело с пятью нотариусами, ни разу проблем не возникло. Важно отметить, что эта квази-проблема в принципе отсутствует, если вы приобретаете дистрибутив на диске: следите только, чтобы при наличии в дистрибутиве проприетарных программ об этом было сказано внятно, т.е. были приведены условия использования. И, конечно, соблюдайте эти условия. Вот передо мной лежит коробочная версия Mandriva Powerpack 2008, выпущенная в оборот компанией «ГНУ/Линуксцентр». На стильной коробке размещена обильная информация о продукте, отдельно упомянуты проприетарные *Cedega* и *Fluendo*. В коробке два DVD (32- и 64-битный), купон с регистрационным номером, на котором среди прочей информации есть указание, что код *Cedega* будет выслан после регистрации, и лицензия с печатью ЗАО «Мезон.ру» на один компьютер; далее, правда, указано, что это ограничение распространяется только на определенную поддержку. (В скобках замечу, что меня эта лицензия во многом не устраивает, но причины этого не относятся к рассматриваемой проблеме.) После инсталляции дистрибутива в нем доступен файл с полной лицензией на русском языке. И что еще нужно?

Естественно, ситуации возникают самые разные: вы скачали продукт для себя либо для продажи, без переработки или с переработкой; вы купили продукт с одной из этих целей; вы, наконец, создаете сборный продукт из разным образом полученных компонентов... И ко всем этим случаям применимо сказанное выше. Для наглядности приведу примеры.

1 Организация приобретает через Интернет проприетарный продукт, скажем, почтовый клиент всех времен и народов *The Bat!* (прекрасно работает в *Wine*, хотя, на мой взгляд, *Claws Mail* – вполне достойная замена). Покупатель скачивает программу, оценивает ее и оплачивает продукт. И получает электронное письмо с кодом, который и регистрирует в соответствующем окне программы. Организация в итоге имеет: счет, платежное поручение с отметкой банка, бухгалтерские проводки, электронное письмо с кодом и, кроме того, если открыть диалог «О программе», в нем будет указано, что владелец экземпляра – именно эта организация.

2 Организация покупает скомпонованный, например, неким ООО диск с набором свободных программ. Возможны три варианта приобретения: через интернет-магазин, у поставщика напрямую и за наличный расчет в розничной сети. Во всех случаях покупатель имеет фирменную упаковку продукта, лицензию продавца со ссылкой на лицензию GNU, бухгалтерские проводки и оригинальный носитель. В первом и втором вариантах он имеет еще счет, платежное поручение с отметкой банка и накладную или иной документ о доставке. В третьем случае он имеет скрепленный с кассовым чеком товарный чек; это неплохо подкрепить приказом о покупке продукта и актом приема-передачи от приобретшего продукт работника.

3 Организация приняла решение скачать продукт из сети. Тут придется немного «помарать бумагу». Первый документ – докладная руководителю о потребности в продукте, с указанием, что он находится в свободном доступе и выпущен под соответствующей лицензией. Уже на этой стадии к докладной можно приложить нотариально удостоверенную копию лицензии (в дальнейшем можно просто ссылаться на нее: «...по той же лицензии, что установленная ранее программа ...»). Второй документ – подробный приказ уполномоченному лицу скачать и установить продукт. Затем последует акт о введении программы в эксплуатацию с указанием, на каком носителе и у кого будет храниться резервная копия. И последнее – продукт должен быть учтен в бухгалтерии. Ко всему этому прибавим содержание окна «О программе» и, скорее всего, наличие текста свободной лицензии в документации.

Заверяю, что с такими наборами организации ничего не грозит.

К вам пришли...

А теперь поговорим о самой неприятной ситуации: на пороге стоит ОБЭП.

Сначала немного статистики. По информации начальника 1-го отдела ОРЧ № 8 УБЭП ГУВД Санкт-Петербурга и Ленинградской области, в 2007 году на обслуживаемой территории возбуждено по ст. 146 УК 199 дел, в суд передано 117 дел. Наиболее суровое наказание получил руководитель торговой сети «505», который провел время до суда в тюрьме. Суд приговорил его к трем годам лишения свободы условно.

ОБЭП может выявлять контрафакт как в процессе обычных проверок финансово-хозяйственной деятельности предприятия, так и во время целевого выхода. Во втором случае это, как правило, обусловлено получением соответствующей информации – от недовольных работников, конкурентов и т.п. В скобках отметим, что для проведения обычных проверок тоже нужна какая-то предварительная информация, но здесь это могут быть и налоговые органы, и контрагенты, и защитники экологии, и прочие. При неразвеянном подозрении на наличие в компьютере пиратских программ он может быть изъят для проведения экспертизы. Надо отметить, что в первую очередь про-

веряющих интересуют «1С», «Гарант», «Кодекс», то есть те компании, которые активно сотрудничают с милицией, даже берут на себя обязательства не заключать с нарушителями соглашений и доводить дело до суда. Теперь в этот список попал и Microsoft.

Сотрудничая с правообладателями, работники милиции знают, во-первых, какие именно продукты они ищут, а во-вторых, признаки, отличающие **данный** легальный продукт от пиратского – пресловутые голограммы, наклейки, оформление коробок и так далее. То есть, наличие или отсутствие такой, например, наклейки явится определяющим признаком только для того продукта, который защищается от нелегального использования с помощью наклейки, и ни для какого иного. Если проще, то на вопрос, а почему у вас нет наклейки Windows, следует ответ: «У меня нет и Windows».

Чтобы определить поведение проверяемого, нужно понять права и возможности работников ОБЭП. И, подчеркиваю, мы говорим о случаях, когда в компьютерах ничего незаконного нет.

Наш интерес вызывает статья 11 Закона «О милиции», повествующая как раз о правах этих стражей правопорядка. И в ней есть два пункта, требующие особого внимания. Это пункт 4, позволяющий *«получать от граждан и должностных лиц необходимые объяснения, сведения, справки, документы и копии с них»*, и пункт 25, описывающий порядок действий *«при наличии данных о влекущем уголовную или административную ответственность нарушении законодательства, регулирующего финансовую, хозяйственную, предпринимательскую и торговую деятельность, в целях установления оснований для возбуждения уголовного дела или дела об административном правонарушении»*. В числе прочего, он позволяет проводить осмотр служебных помещений, изучать документы, отражающие финансовую, хозяйственную, предпринимательскую и торговую деятельность, образцы сырья, продукции и товаров, необходимые для проведения исследований или экспертиз. Важна отметить, что правила этого пункта дают серьезные гарантии проверяемому: так, во-первых, описанные в нем действия проводятся лишь по мотивированному постановлению начальника органа милиции или его заместителя, во-вторых, обязательно присутствие не менее двух понятых и представителя проверяемой организации, в-третьих, все действия протоколируются.

Есть в статье 11 и пункт 35, дающий милиции право *«при наличии достаточных данных, указывающих на признаки преступления, связанного с нарушением законодательства Российской Федерации о налогах и сборах, проводить проверки организаций ...»*, а как уже говорилось, такая проверка может охватить и интересующие нас проблемы. Но, если не вдаваться в юридическую казуистику, права и возможности как проверяющих, так и проверяемых, практически одинаковы при действиях ОБЭП и по пункту 25, и по пункту 35. Поэтому я и сказал о двух позициях, требующих особого внимания – пунктах 4 и 25.

А требуют они внимания потому, что дают лазейку, позволяющую проверяющим превысить свои полномочия. Вот смотрите: приходит к вам работник ОБЭП и говорит, вот, мол, вам письмо, по которому вы, в соответствии с пунктом 4 статьи 11 Закона «О милиции» обязаны сообщить, какое программное обеспечение используете, дать соответствующие объяснения и выдать необходимые справки. И получает огромный массив информации. То есть, он под видом получения сведений проводит незаконную проверку. Повторю, что проведение проверки – значительное вторжение в жизнь предприятия, поэтому это действие формализовано и в законе, и в ведомственных инструкциях в целях соблюдения определенного баланса интересов и возможностей. И оно требует надлежащего процессуального оформления. Закон, кстати, дает работнику милиции право входить беспрепятственно на территорию и в помещения, занимаемые организациями, и осматривать их лишь при преследовании лиц, подозреваемых в совершении преступлений, либо при наличии достаточных оснований полагать, что там совершено или совершается преступление, и в других случаях, не имеющих отношение к рассматриваемым ситуациям – пункт 18 той же статьи 11. Вот работники милиции, допускаю, что из лучших побуждений, будучи не в состоянии законно получить информацию либо реализовать имеющуюся, и подменяют проверку «получением справок».

Ниже я предлагаю определенные алгоритмы поведения. Прошу, однако, помнить, что это именно алгоритмы, а не панацея от всего на свете. Поскольку разговор наш идет все-таки о компаниях, буду исходить из того, что человек с предпринимательским мышлением на уровне своих знаний обладает умением оценивать обстановку и принимать решения, опираясь на знания специалистов. Но именно опираясь, а не отдавая специалистам руководство.

Итак, как стоит действовать

Допустим, компания получает письмо, в котором от нее требуют представить какую-либо информацию. Руководство организации оценивает (в конце концов, и по количественному параметру), запрос ли это конкретной информации или же он «тянет» на полноценную проверку. Во втором случае следует написать вежливое письмо, в котором прямо указать, что по объему и разносторонности запрошенных сведений требование, по мнению организации, выходит за пределы пункта 4 статьи 11. И копию ответа направить в вышестоящий орган милиции, известив об этом и автора запроса. Помните, что на каждого проверяющего есть свой проверяющий. Помните, что за исключением особых случаев (смотрите выше о пунктах 18, 25 и 35 Закона «О милиции»), работник милиции, как и любой другой человек, проходит на предприятие по разрешению его руководителя. На практике от вас или отстают, или вступают в занудную переписку, выливающуюся в приемлемый компромисс. Опыт показывает, кстати, что работникам всякого рода органов предпочтительнее не уступать: если чувствуешь свою правоту, они очень хорошо ощущают твердость и начинают вести себя адекватно, в то время как уступки наводят на мысли о поговорке об отданном пальце.

Другая ситуация. Проверка пришла. Сохраняя хладнокровие, организация знакомится с постановлением; в нем, помимо прочего, персонально перечислены проверяющие. У всех пришедших следует истребовать документы. Если у кого-то документов нет, а про него говорят, что это, мол, специалист по программам, следует это занести в протокол. Понятых следует представить из персонала организации, и не двух, а в количестве, достаточном, чтобы никто из проверяющих не оставался без контроля. К протоколу не надо относиться формально, он должен отражать все действия. Если, например, майор Икс смотрел что-то в комнате А в присутствии понятого Иванова, а капитан Игрек – в комнате Б в присутствии понятого Петрова, это должно быть отражено в протоколе. Ход проверки можно фиксировать на магнитофон или видеозаписью, но делать это открыто, что тоже должно быть занесено в протокол. В протокол должны быть занесены все вопросы, замечания и объяснения. При желании проверяющих изъять компьютер, в протоколе должно быть указано, что организация уверена в отсутствии контрафактных программ, что, по ее мнению, специалист способен разобраться в этом на месте, и, наконец, организация оставляет за собой право взыскать с органа внутренних дел убытки. На практике был случай, когда в аналогичной ситуации в ответ на твердую позицию работники ОБЭП привели какого-то специалиста, он задал два-три вопроса, глянул на компьютер и заявил, что все в порядке.

***

В заключение хочется еще раз сказать, что не надо слепо уповать на юристов, экономистов и других специалистов (из них юристы наиболее активны в предложении своих услуг). Все, о чем мы говорим – это бизнес-ситуации, а не юридические, экономические и прочие. И оценивать их, планировать и принимать решения по ним должен человек, обладающий предпринимательским мышлением, имеющий опыт «распутывания» именно бизнес-ситуаций. Это не значит, что не надо прибегать к услугам специалистов, надо просто ставить перед ними вопросы и задачи в пределах их компетенции. И еще немаловажный момент. Наш мир Open Source очень невелик (надеюсь, пока), и легко найти авторитетного специалиста, например, преподавателя вуза по соответствующей специальности, который в трудную минуту сможет дать мотивированное заключение. LXF

Вопрос — ответ

В этом материале мы постарались рассмотреть типовые вопросы, приходившие на наш редакционный почтовый ящик и задаваемые на различных конференциях и семинарах. Но наверняка что-либо осталось за кадром. Если у вас есть что спросить про использование свободного ПО в России, напишите нам по адресу letters@linuxformat.ru. Мы, в свою очередь, постараемся составить из таких вопросов FAQ, который опубликуем в одном из следующих номеров LXF.

Учебники



Наши эксперты помогут вам с любым приложением Linux



ЕВГЕНИЙ БАЛДИН
Начинал с Агатов. Когда-то даже знал, что такое Робик.

Обучайте примером

Единственный разумный способ обучать людей – это подавать им пример.

Альберт Эйнштейн.

Мне всегда казалось, что чтение хороших публичных лекций – это очень сложно. Так оно и есть, но в некоторых случаях все гораздо проще, чем кажется.

Проверенный заранее компьютер для демонстрационных тестов (желательно ноутбук), любимый дистрибутив и слушатели. Этого достаточно. Как ни удивительно, но даже простую установку и рассказ о ней можно растянуть на пару часов. В конце, отработанным движением руки, можно инсталлировать в Wine программу «Декларация 2007 (Форма 3,4-НДФЛ)» и с чувством глубокого удовлетворения от хорошо выполненной работы на этом закончить. Знакомо? На самом деле такие демонстрации не просто нужны – они необходимы.

GNU/Linux на рабочем столе занимает в районе 1%, и к нему пора привыкать остальным 99. Поэтому берем проверенный ноутбук, любимый дистрибутив и идем искать благодарных, или не очень благодарных, или даже совсем неблагодарных слушателей. Зачем? А чтобы нас стало больше. Да, мы перестанем быть «элитарной группой», правда, не совсем понятно, по какому принципу элитарной. Да, идиотских вопросов прибавится, но и умных людей в нашем окружении – тоже. Не таких как мы – это и есть самое главное. А вот для них и с их помощью можно делать и более вменяемые презентации.

E.M.Baldin@inp.nsk.su

В этом выпуске...



Совет месяца: Webmin



Текстовые конфигурационные файлы – это хорошо: работать с ними куда проще и быстрее, чем бегать по нескольким уровням вложенности в GUI, при условии, конечно, что вы знаете, как опции в каком файле надо менять. Потому-то многие дистрибутивы и предлагают графические мастера настройки: они делают жизнь не-гуру проще.

Но у мастеров есть и недостатки. Они работают лишь с одним дистрибутивом, обычно требуют наличия рабочего стола и поддерживают лишь подмножество опций, предусмотренное их авторами. А что если вы настраиваете удаленный сервер или делаете нечто, о чем не подумали создатели дистрибутива? А как администрировать несколько машин?

Ответ на все эти, а также многие другие вопросы – *Webmin* (www.webmin.com). Это инструмент настройки через браузер, написанный на Perl и включающий собственный web-сервер. Следовательно, он может работать почти на любой системе – и нет нужды мучаться с *Apache*. Установив пакет, откройте в браузере адрес https://localhost:10000, и посмотрите, что умеет *Webmin*.

Если вы пытаетесь получить доступ к *Webmin* с другого компьютера (в локальной сети или через Интернет), а настройки дистрибутива по умолчанию блокируют входящие соединения – откройте его локально, в Links через SSH, и разрешите подключение в соответствующем разделе.

Gmail: Пересылка

Запустить *Thunderbird* в ваш web-ящик – не самое простое занятие, но **Энди Ченнел** покажет: стоит разок помучиться – и революция в приеме почты обеспечена!



Наш эксперт

Энди Ченнел
Энди делает свои первые шаги в Linux уже шесть лет, а технологиями интересуется со времен Dragon 32.

В мире есть сотни провайдеров web-почты, и большинство из них предоставляют POP3- и IMAP-доступ к вашему ящику. На данном уроке мы настроим *Thunderbird*, почтовый клиент Mozilla, для работы с Gmail и по POP (Post Office Protocol – почтовый протокол), и по IMAP (Internet Message Access Protocol – протокол доступа к сообщениям в Интернете), а также приведем несколько хитрых приемов, которые здорово упростят организацию вашей корреспонденции.

Для этого занятия мы взяли Gmail, решение от Google, но все должно быть переносимым на любую другую web-почту. Спросите, зачем вам использовать ее вместо почтового ящика от вашего интернет-провайдера? Во-первых, тогда смена провайдера не означает смену адреса или потерю архива почты. Во-вторых, большинство интернет-провайдеров ограничивают хранилище почты объемом 100 МБ – вроде и много, но иногда приходится удалять что-то или терять новые сообщения; а вот учетной записи Gmail обеспечено 6 ГБ, и это число постоянно растет. На практике это означает, что если вы послали письмо с вложением в **LXF UK** после 8 сентября 2004 года, то оно все еще у нас есть, поскольку объем хранилища (и поиск) позволяет ничего не удалять; отправленные сообщения также сохраняются. Так что у нас есть полный архив переписки за последние 3,5 года. Здорово!

А главное, применение web-почты совместно с локальным почтовым клиентом вроде *Thunderbird* означает использование преимуществ обоих миров: вы имеете полный контроль над вашей почтовой системой дома, но можете также получить доступ к web-ящику, находясь в отъезде.

Web-доставка

Мы начнем с настройки доступа *Thunderbird* к вашей почтовой записи. Gmail предлагает простую пересылку и доступ по POP3 или IMAP; POP3 шире распространен, а IMAP предоставляет больший контроль над службой посредством двунаправленной синхронизации. То есть, если вы создаете папку с именем «Рыбалка» на вашей локальной машине, то такая же папка будет создана и на сервере. Аналогично, если вы удаляете сообщения с IMAP-сервера, они будут уничтожены и локально. С более практичной точки зрения, с помощью IMAP и сервер, и клиент отслеживают, прочитали ли вы конкретное сообщение, тогда как POP – строго односторонняя служба – нет.

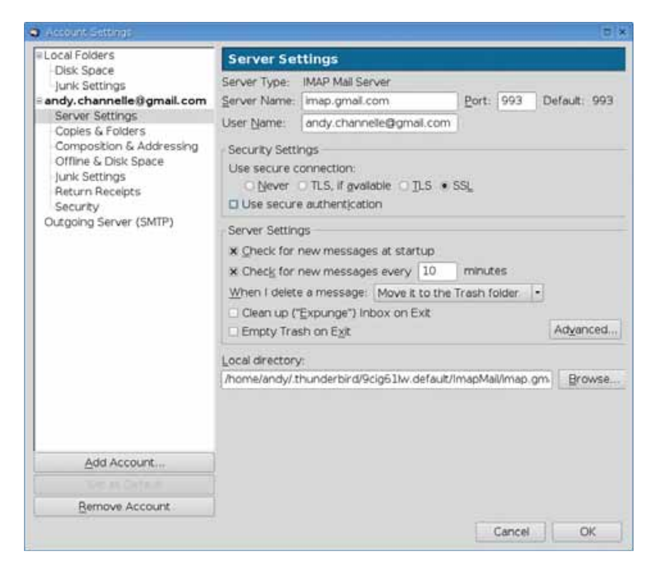
Для целей нашего урока предположим, что у вас есть адрес, предоставленный вашим провайдером (например, bob@bob.net), вы настроили *Thunderbird* на эту учетную запись, и имеете еще один почтовый ящик на Google, то есть второй адрес (например, bob@gmail.com). Это идеальная ситуация для настройки пересылки, и поскольку главное слово здесь «пересылка», то большая часть работы происходит с настройками Gmail, так что войдите под своей учетной записью и щелкните на ссылке Настройки вверху справа. Нажмите на вкладке Пересылка и POP/IMAP, щелкните на радиокнопке с надписью «Пересылать копии входящих писем на...» и введите свой электронный адрес в текстовое поле. Вы также можете настроить сохранение копий в почтовом ящике Google, архивирование или удаление после отсылки сообщения; и поскольку это ничего не стоит, мы советуем сохранять их.

Этот способ управления почтой превосходен, когда вы переходите с учетной записи вашего провайдера на web-почту: сообщения, предназначенные bob@bob.net, все еще будут приходить, тогда как вы будете использовать свой новый адрес bob@gmail.com. Все ваши сообщения будут прибывать в одно место, и вы сможете настроить *Thunderbird* так, чтобы письма отправлялись с bob@gmail.com, при помощи Мастера учетной записи.

Следующий шаг – вообще забыть об учетной записи у провайдера и настроить *Thunderbird* на прямое подключение к Gmail. И вновь это подразумевает обращение к настройкам Gmail, но большая часть работы выполняется в *Thunderbird*. Итак, в Gmail выберите Настройки > Пересылка и POP/IMAP, выберите Включить POP для писем, полученных с данного момента, а затем укажите Google, как обращаться с сообщениями после их загрузки. Не забудьте нажать кнопку Сохранить изменения; теперь можно перейти к почтовому клиенту.

В почтовой учетной записи есть два основных элемента – входящий и исходящий серверы; но для начала надо создать ящик: мастер *Thunderbird* проведет нас через этот процесс. При запуске Мастера вы заметите, что имеется выбор Учетная запись электронной почты и Gmail; мы укажем последнее. В следующем окне следует ввести ваше (настоящее) имя и название учетной записи Gmail – все символы до @gmail.com. Затем нажмите Далее, подтвердите информацию, нажав Завершить, и мы готовы к приему почты.

Для служб, отличных от Gmail, начало такое же, но придется добавить информацию о серверах POP и SMTP (Simple Message Transfer Protocol – простой протокол передачи почты); эти настройки можно



› **IMAP допускает двустороннее соединение с почтовым сервером, что для большинства пользователей полезнее, чем POP.**

» **Месяц назад** Мы проникли в тайны KDE4 и завоевали *Konqueror*.

файлов и настройка

найти, воспользовавшись интерфейсом поставщика web-почты, если он предлагает POP-сервис. Определив POP/SMTP-службы, вы сможете отправлять сообщения из *Thunderbird*, и они будут пересылаться через SMTP-сервер Goggle, а сообщения, поступающие на ваш почтовый ящик Gmail, будет периодически загружаться для локального прочтения. Однако POP-доступ не окажет никакого влияния на саму учетную запись Gmail – то есть если вы не принимали почту всю неделю, то накопится достаточно большое количество «непрочитанных» сообщений (которые вы уже изучили локально), а это слегка утомляет, когда вы вновь заходите в web-интерфейс.

Магия IMAP

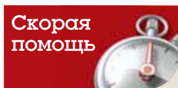
Решение – использовать IMAP вместо POP. Так что вернитесь в Gmail > Настройки > Пересылка... и выберите Включить IMAP на третьей вкладке. Теперь вернитесь в *Thunderbird* и перезапустите Мастер учетной записи. Выберите из списка Учетная запись электронной почты, нажмите Далее и укажите IMAP как тип входящего сервера. Имя сервера входящей почты будет **imap.gmail.com**, а исходящей – **smtp.gmail.com**. Нажмите Далее и добавьте необходимое имя пользователя для входящей и исходящей почты. Это ваше полное имя пользователя в Gmail, то есть **bob@gmail.com**. В следующем окне, на левой панели выберите Настройки сервера и добавьте следующую информацию. Порт сервера должен быть 993, настройки безопасности следует установить в SSL. В нижней половине окна раздела настроек почтового сервера выбирается интервал проверки почты. Если вы подключаетесь с телефонной линии, помните, что настройка на 10 минут означает, что каждые 10 минут *Thunderbird* будет пытаться установить соединение и загрузить почту.

Теперь выберите опцию Исходящий сервер (SMTP) и добавьте/измените следующее: имя сервера должно быть **smtp.gmail.com**, порт сервера – 587, а в разделе Защита и аутентификация следует ввести свое полное имя пользователя и выбрать безопасное соединение TLS. Нажмите OK, и вы будете готовы к действиям, хотя при первом соединении у вас запросят пароль. Если вы – пользователь Gmail со стажем и накопили достаточно писем, то это первое соединение может занять много времени, поскольку потребуется загрузить на вашу машину большой объем информации; регулярные подключения должны быть более быстрыми.

Одной из причин, почему мы выбрали для данного учебника Gmail, а не другой сервис, является доступность нескольких уловок для облегчения жизни. Например, можно сортировать входящую почту еще до того, как она будет отослана. Это выполняется добавлением аргумента **+** или суффикса к вашему имени пользователя. Например, пусть вы подписались на пять рассылок о кошках. В обычной ситуации вам пришлось бы настроить пять фильтров в Gmail, чтобы обеспечить перенос сообщений каждой из рассылок в нужную папку. Вместо этого вы можете использовать **bob+cats@gmail.com** (**+cats** – это аргумент), а затем настроить один фильтр на основе этого входящего адреса, и все автоматически будет отправляться/помечаться как «Cats».

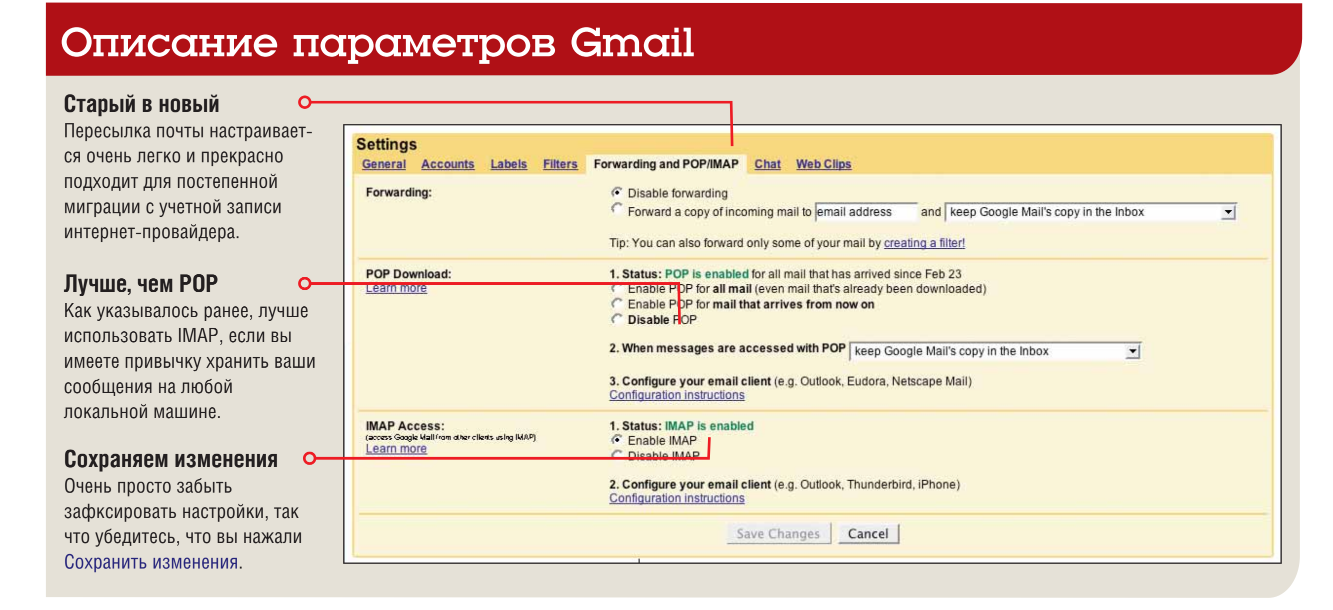
При большом количестве писем вы можете пропустить что-то важное. Все ваши непрочитанные сообщения можно найти, создав ярлык с именем Unread (перейдите в Настройки, выберите Ярлыки и добавьте имя в поле Создать новый ярлык). Теперь при выборе этой метки в левой части интерфейса пользователя Gmail вы будете видеть только непрочитанные сообщения.

Gmail весьма либерален к формату имени пользователя. Если ваш адрес **bob.johnson@gmail.com**, то любые варианты **bob.johnson** также будут работать. Например, можно использовать **bob.johnson**, **bobjohnson**, **bobjohn.son** и так далее – все они действительны. Это прекрасно, если у вас плохая память или если вы собираетесь подписаться на различные службы и хотите фильтровать или помечать сообщения автоматически, как в первом совете.



Если у вас более одной учетной записи Gmail, загрузите *Gmail Manager for Firefox*. **https://addons.mozilla.org/firefox/1320/**

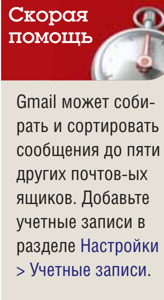
Описание параметров Gmail



Старый в новый
Пересылка почты настраивается очень легко и прекрасно подходит для постепенной миграции с учетной записи интернет-провайдера.

Лучше, чем POP
Как указывалось ранее, лучше использовать IMAP, если вы имеете привычку хранить ваши сообщения на любой локальной машине.

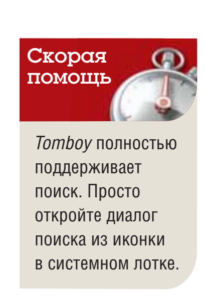
Сохраняем изменения
Очень просто забыть зафксировать настройки, так что убедитесь, что вы нажали Сохранить изменения.



Gmail может собирать и сортировать сообщения до пяти других почтов-ых ящиков. Добавьте учетные записи в разделе Настройки > Учетные записи.

Tomboy и Tasque:

Хранение всей информации под рукой весьма экономит время и для работы, и на личные нужды. Может, хоть теперь **Энди Ченнел** будет сдавать статьи в срок?



Tomboy полностью поддерживает поиск. Просто откройте диалог поиска из иконки в системном лотке.

Не исключено, что ваш рабочий стол облеплен маленькими желтыми бумажками-напоминаниями. Иногда этот метод организации даже работает, но если вы утонули в Post-It™-клочках, то почему бы не перенести все эти факты, цифры и идеи в ваш компьютер? Воспользовавшись приложением *Tomboy*, мы создадим заметки, увяжем их вместе, и, что более важно, приспособим их для поиска уже введенной информации. Мы также представим новейшее приложение *Tasque*, которое изумительно интегрируется с Remember The Milk [Не забудь молоко] (зарегистрируйтесь бесплатно на **www.rememberthemilk.com**), сетевым ПО для управления списком задач – куда бы вы ни уехали, ваши дела догонят вас через Интернет. Причина использовать их вместе именно сейчас – в том, что они будут интегрированы уже в следующей версии, так что, например, задачи *Tasque* можно будет создавать прямо в *Tomboy*.

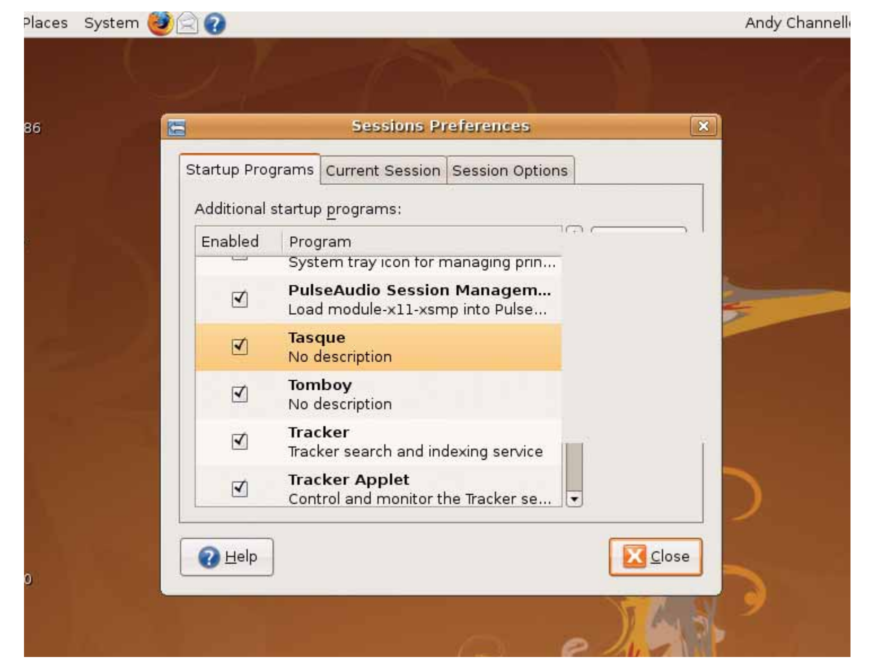
И *Tomboy*, и *Tasque* – это проекты, основанные на Mono и рожденные в лабораториях Novell. *Tomboy* – вполне зрелое приложение, доступное в большинстве распространенных дистрибутивов Linux; например, Hardy Heron, последний релиз Ubuntu, поставляется с версией 0.10, являющейся текущим стабильным релизом. А вот *Tasque* только что вышло в свет, и потому требует небольших усилий при установке. Пользователям SUSE 10.3 легче всего, поскольку программа доступна для установки «одним щелчком» на **http://live.gnome.org/Tasque/Building**. Пользователи Ubuntu могут получить двоичный файл, открыв Synaptic (Система > Администрирование > Менеджер пакетов Synaptic, перейдя в Настройки > Репозитории, щелкнув на вкладке Сторонние приложения и добавив следующий репозиторий:

```
http://ppa.launchpad.net/tasque-packagers/ubuntu hardy main
```

или

```
deb http://ppa.launchpad.net/tasque-packagers/ubuntu gutsy main
```

После этого нажмите кнопку Поиск и введите **Tasque**; появится приложение, готовое к установке. Щелкните на иконе слева и выберите



› Добавьте *Tomboy* и Tasque к вашим программам автозапуска, чтобы они были постоянно доступны при загрузке.



› **Необходима помощь в проверке правописания? Щелкните на иконке *Tomboy*, выберите Настройки и убедитесь, что на вкладке Редактирование включена проверка орфографии.**

Пометить для установки, затем нажмите кнопку Применить.

Заметки и напоминания

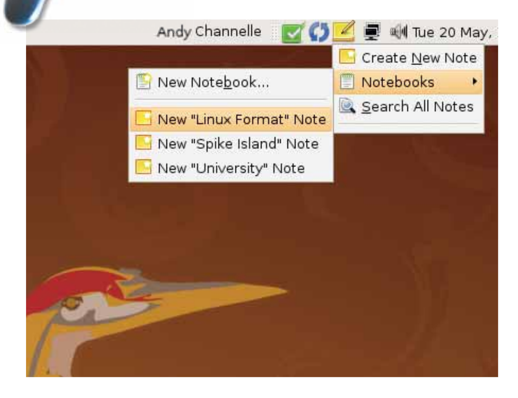
Tomboy работает как очень простой web-сайт Wiki, что делает его полезным для создания серии связанных заметок с возможностью поиска, индексирования и опроса различными способами. Более того, если вы писали статьи в Википедию и использовали WikiWords (два слова, написанные слитно, каждое с заглавной буквы), то можете использовать этот же формат и для создания и связывания новых страниц: нужно только отметить опцию WikiWords. *Tomboy* и *Tasque* разрабатывались так, чтоб всегда быть под рукой, поэтому при первом запуске каждого приложения они помещаются в системном лотке вашего рабочего стола (в Ubuntu он у правого края экрана), чтобы их основные элементы были доступны в одно нажатие. Для большего удобства можно настроить запуск *Tomboy* и *Tasque* при загрузке, чтобы они всегда присутствовали в системном лотке, как показано на экранном снимке внизу слева. Для этого выберите Система > Настройки > Сессии и добавьте в список новую запись. Введите затребованную информацию: команда для *Tomboy* – это **tomboy**, а для *Tasque* наберите **tasque**; затем нажмите кнопку Закрыть. При следующей загрузке оба апплета будут готовы к использованию.

Готовность к старту

После запуска *Tomboy* будет заполнен парой заметок (Начните отсюда и Создание ссылок в Tomboy). А нам нужен чистый список, так что выберите каждую из них в меню *Tomboy* – щелкните на иконке в системном лотке для просмотра заметок – и нажмите кнопку Удалить. Подтвердите свои намерения, и заметки исчезнут. Если мы просто приступим к созданию заметок, то, вероятно, через несколько дней их список станет такой же свалкой, что и лес Post-it™, так что лучше начать с нескольких «записных книжек»: их можно использовать для структурирования. Чтобы создать новую записную книжку, щелкните на иконке *Tomboy*, выберите Записная книжка > Создать записную книжку и введите имя. После нажатия кнопки Создать, ваша новая записная книжка будет добавлена в список, и вы сможете создавать в ней новые заметки из интерфейса программы.

дела в порядке!

Шаг за шагом: Организуйте свою жизнь

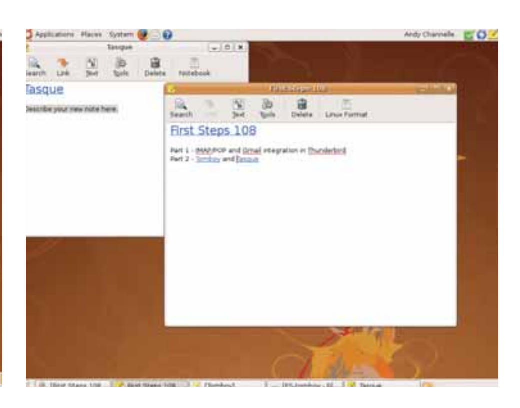


1 Создаем новую заметку

Создайте новую заметку, щелкнув на иконке *Tomboy*, выбрав Записные книжки и затем указав Создать «что-то», где что-то – это имя вашей записной книжки. Откроется новая заметка, готовая к заполнению информацией. Первым делом надо задать ей заголовок, выделив текст в самом верху страницы и введя новый. В будущем на него будет вести ссылка.

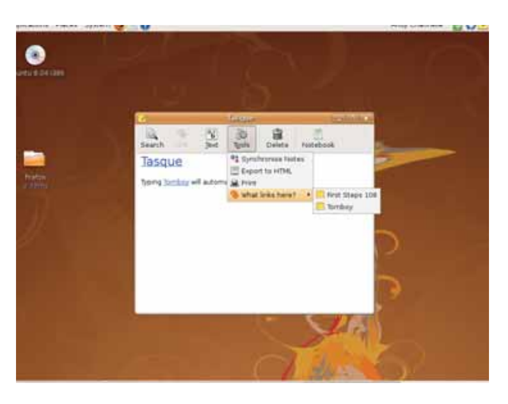
2 Добавляем текст

Выделите стандартный текст и введите то, что желаете. Можете форматировать слова и создавать маркированные списки, используя меню Текст. Хотя эти опции не посрамят *OpenOffice.org*, их достаточно для создания довольно сложных заметок. В верхней строке вы заметите, что по умолчанию новая заметка не присоединена ни к одной записной книжке. Щелкните на иконке записной книжки и выберите соответствующее имя, чтобы это исправить.



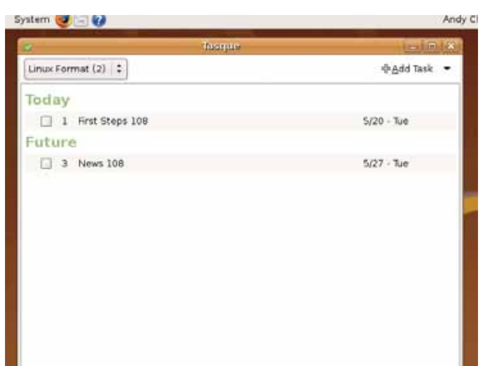
3 Связываем заметки

Именно ссылки отличают *Tomboy* от конкурентов. Выделите часть текста заметки и нажмите кнопку Ссылка (или щелкните правой кнопкой мыши и выберите Ссылка на новую заметку). Волшебным образом появится новая заметка, готовая к вводу текста. Единственное, к чему придется привыкать – это к факту отсутствия кнопки Сохранить; все записывается по мере набора благодаря оболочке базы данных *Tomboy*.



4 Автоматические ссылки

Теперь – действительно крутой трюк. Допустим, вы создаете заметку с именем День спорта и ссылку от нее к заметке с именем Победители (как было указано выше). Когда в заметке Победители вы вводите словосочетание «День спорта», то оно автоматически становится ссылкой на исходную заметку, и щелчок мышью откроет ее. Вы можете увидеть, что ссылается на заметку, открыв ее и выполнив Инструменты > Какие ссылки здесь?



5 Tasque в руках

Теперь перейдем к *Tasque*. Если вы подписались на Remember the Milk, настройте его как «службу», чтобы использовать в *Tasque*. Как только вы авторизуете приложение в RTM, в выпадающем списке *Tasque* должны автоматически появиться ваши списки задач, с числом просроченных дел напротив каждого из них. Вы можете добавить задачу, выбрав соответствующий список и нажав кнопку Добавить.

6 Расставляем приоритеты

Задачам может быть присвоен приоритет (1 – наиболее важный) и дата завершения, путем щелчка слева и справа списка соответственно, и они будут показываться, по умолчанию, в зависимости от даты завершения. Щелчок на радиокнопке удалит задачу из списка, но вы все еще сможете видеть выполненные задачи, перейдя в Настройки (правый щелчок на иконке в системном лотке) и выбрав Показать выполненные задачи. LXF

Через месяц Мы переместимся в сеть и посмотрим, что могут предложить Flock и Google Desktop.

Eee PC: Добавьте

Часть 2: Ваш Eee не ограничен одним дистрибутивом. **Майк Сондерс** покажет, как установить другой вариант Linux и зарезервировать файлы.



Наш эксперт

Майк Сондерс прикупил Eee PC еще в ноябре, и сделал все мыслимые и немыслимые настройки. Он носится со своим ноутбуком как с маленьким пластмассовым ребеночком.

Если вы прочитали тему номера в LXF106 и следовали уроку прошлого месяца, то уже достаточно освоились с механикой рабочего стола вашего Eee: мы добавляли новые приложения и подстраивали интерфейс. Но если вы только-только прикупили Eee и не видели предыдущих статей, не расстраивайтесь – данный урок самодостаточен, и проделывать описанные ранее изменения незачем. Приятнее всего то, что можно не беспокоиться о повреждении настроек вашего Eee: чуть позже мы покажем, как сделать резервную копию каждого файла на вашей машине одним кавалерийским наскоком!

Дистрибутив на базе Xandros, которым Eee снабжается по умолчанию, превосходен для офисных работ и просмотра Интернета, и вы можете усилить его дополнительными пакетами из репозиториев Xandros. Но это не есть все и вся в Linux: как ни мал Eee, это полноценный ПК, и на нем способны запускаться и другие дистрибутивы. В некоторых случаях оборудование Eee потребует дополнительных драйверов, которые обычно не представлены в настольных системах, но к нашей радости, в мире есть достаточно дистрибутивов, подстроенных специально для Eee.

На этом уроке мы покажем, как установить на Eee дополнительную систему. Но зачем бы вам это понадобилось?

» **Безопасность**. Если Xandros не загрузится, вы сможете запустить второй дистрибутив и исправить это.

» **Игры** Удивительно, но Eee имеет мощность небольшой игровой приставки Windows, даже *GTA: Vice City* идет очень хорошо. Если вы захотите установить Windows как основную ОС, вы опять-таки сможете легко вернуться в Linux.

» **Интерес** Пробовать новые дистрибутивы и испытывать больше разновидностей Linux – всегда пленительный процесс.

Для урока вам потребуется SD-карта объемом не менее 512 МБ, которую вы сможете подключить к вашему Eee (слот с правой стороны). Они сейчас поразительно дешевы: 2-ГБ карту можно достать через Интернет за 10-15 долларов.

Часть 1 Добавление нового дистрибутива

Давайте уже начнем. Мы собираемся установить новый дистрибутив Linux на SD-карту, что даст вам возможность выбирать между им и Xandros. Мы возьмем Breeezy (http://tinyurl.com/2e36u6), специальную версию мини-дистрибутива Puppy Linux, подстроенную под оборудование Eee. Когда мы закончим, вы сможете вызвать загрузочное меню, где можно будет выбирать из Xandros с внутреннего накопителя или Breeezy с SD-карты. Нам предстоит немного поработать в командной строке – если это приводит вас в ужас, прочитайте руководство Новичку в Linux из раздела Справка на **LXFDVD**, и вы почувствуете себя намного увереннее.

Загрузите ваш Eee без вставленной SD-карты, и если у вас нет способа взять данные с **LXFDVD** (например, нет внешнего DVD-привода), соединитесь с Интернетом. Вот те файлы, что будут использованы:

» **breeezy.zip**

» **syslinux-nomtools**

Вы найдете их в разделе **Distros/Breeezy** на DVD или на www.linuxformat.ru/download/breeezy. В любом случае, сохраните эти файлы в ваш домашний каталог на Eee и следуйте нашим пошаговым инструкциям [Breeezy занимает 65 МБ. Если у вас дорогой трафик и нет внешнего DVD, просто скопируйте его на Eee через ПК, воспользовавшись USB-брелком, – прим.ред.].

Приготовьте SD-карту

Вставьте вашу SD-карту, и если появляется окно Removable Device, просто нажмите Cancel – нам не надо просматривать ее содержимое в файловом менеджере. Теперь откройте окно командной строки, нажав Ctrl+Alt+T. Введите **ls** для отображения содержимого вашего домашнего каталога, и, наряду с файлами **breeezy.zip** и **syslinux-nomtools**, которые были скопированы ранее, вы также должны увидеть каталог для SD-карты под именем **MMC-CD**.

Сначала нам нужно переформатировать SD-карту, чтобы она содержала файловую систему, пригодную для загрузки, под названием FAT16. (Сделайте резервные копии всех важных данных на вашей SD-карте – мы собираемся все там стереть!) Введите **sudo bash**, чтобы переключиться на пользователя root (администратора), затем введите эти команды:

» **Месяц назад** Добавление новых приложений и включение рабочего стола KDE.

дистрибутивов




```
user@eeepc-mike
Partition number (1-4): 1
First cylinder (1-1018, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-1018, default 1018):
Using default value 1018

Command (m for help): t
Selected partition 1
Hex code (type L to list codes): e
Changed system type of partition 1 to e (W95 FAT16 (LBA))

Command (m for help): a
Partition number (1-4): 1

Command (m for help): p

Disk /dev/sdb: 509 MB, 509083648 bytes
16 heads, 61 sectors/track, 1018 cylinders
Units = cylinders of 976 * 512 = 499712 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1   *           1        1018      496753+   e  W95 FAT16 (LBA)

Command (m for help):
```

› Чтобы создать на SD-карте новый раздел для Breeezy, используйте *fdisk*.

```
umount /dev/sdb1
fdisk /dev/sdb
```

SD-карта отмонтируется (это даст нам неограниченный доступ к ней); затем запустится менеджер разбивки диска. В приглашении введите **o**, для создания пустой таблицы разделов. Затем введите **n** (новый раздел), потом **p** (первичный раздел) и **1** на следующем шаге. Тут вас попросят указать, какие цилиндры диска использовать – просто дважды нажмите Enter, чтобы принять умолчания. Наконец, введите **p**, чтобы посмотреть сведения о новом разделе.

Нам нужно сменить тип нового раздела на FAT16: введите **t**, и когда у вас спросят тип, ответьте **e**. Также нам нужно сделать раздел загрузочным: введите **a** (активировать), затем **1** на следующий вопрос. Теперь для отображения нового раздела можно ввести **p** еще раз, и ваш результат должен походить на верхний снимок экрана. Наконец, введите **w**, чтобы записать изменения на SD-карту и выйти.

Для создания на ней новой файловой системы, введите

```
mkfs.vfat /dev/sdb1
```

Теперь наша SD-карта полностью готова, и можно добавлять файлы дистрибутива Breeezy. Введите **exit**, чтобы вернуться обратно в режим обычного пользователя.

Установка Breeezy

Вытащите SD-карту и вставьте ее, чтобы она примонтировалась снова. (Как и раньше, щелкните по Cancel при возникновении диалога менеджера устройств.) В командной строке введите:

```
cd MMC-SD/partition1/
```

Теперь мы находимся в главном каталоге SD-карты, и можем извлечь файлы Breeezy примерно так:

```
unzip ~/breeezy.zip
```

(Вы найдете символ ~, нажав Shift+ё.) Не волнуйтесь, если появились предупреждения «file attributes error» – это из-за того, что вы извлекаете файлы Linux на SD-карту, отформатированную под FAT16. Введите **ls -lh**, чтобы увидеть новые файлы Breeezy, как на экранном снимке.

Почти готово; осталось просто добавить загрузчик *Syslinux*. Введите **cd** (сам по себе), чтобы переместиться обратно в домашний каталог, а затем **sudo bash**, чтобы снова оказаться в режиме root. Теперь введите:

```
umount /dev/sdb1
chmod +x syslinux-nomtools
./syslinux-nomtools /dev/sdb1
```

Перезагрузите ваш Eee со вставленной SD-картой. В Asus'овском экране загрузки непрерывно жмите Esc, пока не попадете в загрузочное меню. Вы увидите два пункта: первый для Xandros, ОС внутреннего накопителя, а второй (CardReader SD0) для SD-карты с Breeezy.

Использование Breeezy

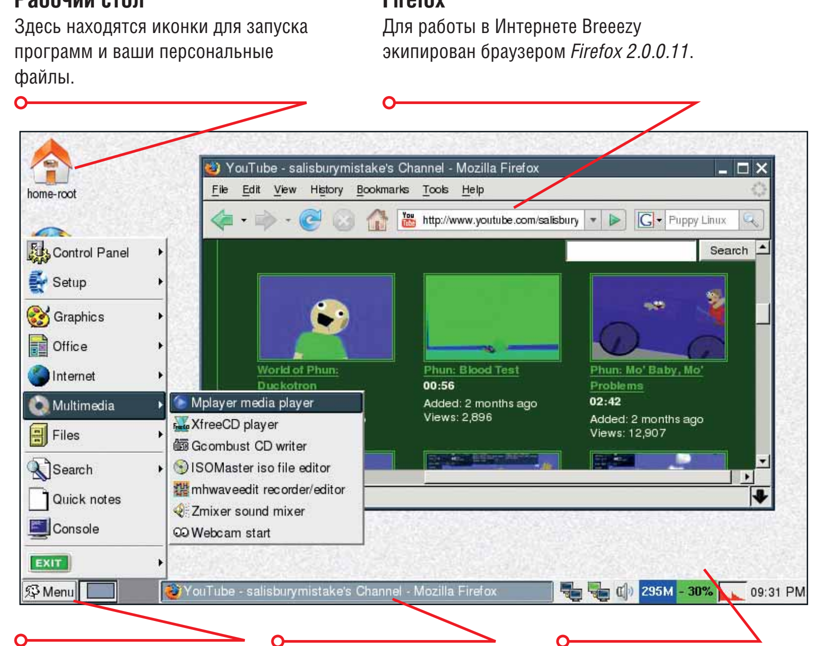
Примерно минуту спустя вы прибудете на рабочий стол Breeezy, и услышите из колонок тявкающий звук (ведь Breeezy базируется на Puppy Linux, а puppy по-англ. «щенок»). Выскочит web-браузер *Dillo*, с советами и подсказками по использованию дистрибутива – ознакомьтесь с ними. Вы заметите, что сначала будет отсутствовать панель задач – переместите мышь вниз экрана, и она появится. Все это во многом похоже на *IceWM*: меню программ слева, виртуальные рабочие столы и панель задач внизу, а кое-какая статистика (память и батарея) – справа.

Щелкните по кнопке Menu, чтобы открылся главный список имеющихся приложений. Помните, что родительский дистрибутив для Breeezy, Puppy Linux, разрабатывался для старых машин (например, Pentium 166), и на значительно более мощной технике, такой как Eee, Breeezy просто летает. Программы в его составе очень быстры и легки – взгляните на графический редактор *MTPaint* из раздела Graphics и *TextMaker* в секции Office для работы с документами. »

Исследуем рабочий стол Breeezy

Рабочий стол
Здесь находятся иконки для запуска программ и ваши персональные файлы.

Firefox
Для работы в Интернете Breeezy экипирован браузером *Firefox 2.0.0.11*.



Меню
Щелкните по этой кнопке, чтобы выпал список имеющихся приложений.

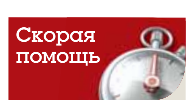
Панель задач
Переместите указатель мыши в низ экрана, чтобы она стала видимой.

Обратная связь
Синее число показывает доступную память, а зеленое – заряд батареи.

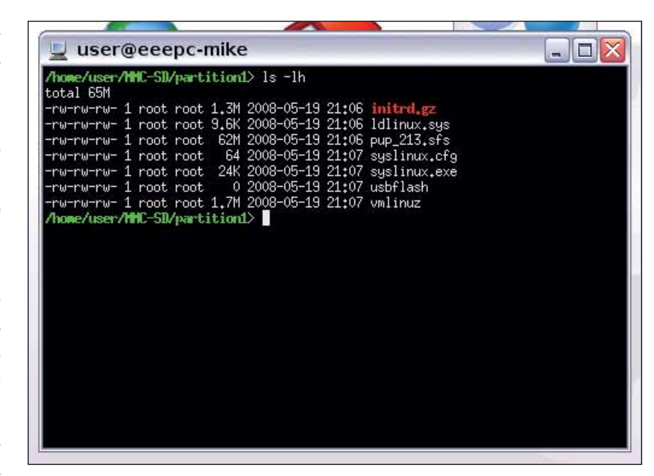
Для исследования Интернета Breeezy содержит два браузера: упомянутый выше *Dillo* и *Firefox*. *Dillo* очень быстрый, но не поддерживает многие web-функции – попытавшись использовать его для сайтов, нашпигованных JavaScript, вы недалеко уйдете. Для поиска в Google и блуждания в лабиринтах данных Википедии *Dillo* вполне годится, но для большинства задач стоит использовать *Firefox*.

Breeezy хорошо оснащен и по части мультимедиа: попробуйте *MPlayer* для воспроизведения многочисленных форматов видео и аудио и *MHWaveEdit* для записи звука с микрофонного входа. Чтобы настроить Breeezy, перейдите в меню Setup, а оттуда – в сеть через мастер Network. Соединение с wi-fi весьма просто – щелкните Wireless в главном окне настройки сети и Scan внизу для выбора сети. Все, что нужно будет вам далее – это щелкнуть по Use This Profile и Auto DHCP для соединения. Теперь запускайте *Firefox*.

Breeezy – отличный небольшой дистрибутив и хорошая альтернатива Xandros OS, стоящему по умолчанию. Если вы планируете установить на внутренний накопитель другой дистрибутив (а то и Windows, а может, она была там с самого начала?), вы все еще можете держать Linux под рукой, а точнее – Breeezy на вашей SD-карте. Вы даже можете добавить приложений посредством Menu > Setup > Full package manager, если вы в Сети. Чтобы выключить Breeezy, перейдите в меню Exit – вас попросят сохранить изменения.



В Breeezy, как и в Xandros, вы можете двигать окна размера большего, чем экран, путем удерживания Alt, щелчка и перетаскивания Это полезно для программ, которые слишком велики для мизерного Eee'шного разрешения 800х480.



› Разархивировав Breeezy, введите ls -lh, для вывода этого списка файлов дистрибутива на SD-карте.

Часть 2 Создание резервных копий на Eee

Часто резервирование – это кропотливая работа, именно поэтому никто из нас не делает ее так часто, как следовало бы. Какие файлы следует сохранять? Следует ли резервировать скрипты загрузки? Хватит ли места на DVD-R? Флэш-память на Eee – 4 ГБ; конечно, не развернешься, зато вы сможете делать несколько резервных копий всего вашего Eee, не потребляя слишком много места. Каждая из них будет занимать не больше, чем 4 ГБ, а со сжатием – и того меньше.

Возможность делать несколько резервных копий очень удобна. Например, на нашем тестовом Eee мы сделали резервные копии для исходного дистрибутива Xandros, перенастроенной версии Xandros и установки Syllable (www.syllable.org). Когда вы оборудуете Eee всем, что душеньке угодно, с идеальным выбором программ и настроек, можете резервировать всю машину, на случай, если будущее обновление или установка приложения испортит дело. Это также означает, что можно смело пробовать другие дистрибутивы и ОС, восстанавливая Eee к первоначальному состоянию при желании вернуться к нормальному положению дел.

Мы сделаем это, создав точный клон – бит-в-бит – внутреннего флэш-накопителя Eee. Вместо того, чтобы соображать, какие файлы ОС нам нужны и как их разместить, просто скопируем все подряд в одну большую кучу. Вам потребуется какое-либо устройство хранения; на данном уроке мы используем внешний USB-диск, но если у вас есть USB-брелок с 4 ГБ или больше свободного места, это тоже сработает. Установите Breeezy, как написано в первой части урока, загрузите его и следуйте дальнейшим шагам.

Добавление внешнего накопителя

Breeezy не автомонтирует USB-накопители, но это нетрудно сделать вручную. Вставьте ваш жесткий диск или флэш-брелок и щелкните по кнопке Menu и Console, чтобы запустить окно командной строки. Затем введите

```
dmesg
```

Появится список сообщений ядра, большая часть которых нам не потребуется. Но взглянув на последние несколько строк, вы увидите информацию о накопителе, который только что подключили. Отыщите что-нибудь вроде этого:

```
SCSI device sdb: 78140160 512-byte hdwr sectors (40008 MB)
sdb: Write Protect is off
sdb: Mode Sense: 53 00 00 08
```

Вы увидите множество других строк, начинающихся с sdb, или, возможно, sdc, в зависимости от того, вставляли ли вы до этого другой накопитель – в качестве примера взгляните на снимок экрана. Самая важная строка сообщает нам, сколько разделов имеется на устройстве: она выглядит примерно так:

```
sdb: sdb1
```

(Обратите внимание на отступ в один пробел.) Отсюда мы сможем определить, что нашему внешнему накопителю присвоено имя устройства sdb, и он имеет один раздел под названием sdb1. Мы добавим резервную копию Eee на sdb1, но если у вас есть второй раздел, где больше места, можете использовать sdb2.

```
rxvt
wifi0: Atheros 5424/2424: mem=0xfbef0000, irq=10
ACPI: Battery Slot [BAT0] (battery present)
Asus EEEPC HOTKEY
[eeepc hotk] Hotkey init flags 0x41.
[eeepc hotk] Get control methods supported : 0x101711
usb 5-2: new high speed USB device using ehci_hcd and address 3
usb 5-2: configuration #1 chosen from 1 choice
scsi2 : SCSI emulation for USB Mass Storage devices
usb-storage: device found at 3
usb-storage: waiting for device to settle before scanning
  Vendor: SEAGATE   Model: ST9402115A        Rev: 3.01
  Type:   Direct-Access                      ANSI SCSI revision: 02
SCSI device sdb: 78140160 512-byte hdwr sectors (40008 MB)
sdb: Write Protect is off
sdb: Mode Sense: 53 00 00 08
sdb: assuming drive cache: write through
SCSI device sdb: 78140160 512-byte hdwr sectors (40008 MB)
sdb: Write Protect is off
sdb: Mode Sense: 53 00 00 08
sdb: assuming drive cache: write through
 sdb: sdb1
sd 2:0:0:0: Attached scsi disk sdb
usb-storage: device scan complete
sh-3.00#
```

› Используя команду dmesg, мы можем идентифицировать имя устройства для внешнего накопителя, чтобы его можно было примонтировать.

Монтирование накопителя

Теперь нам нужно примонтировать внешний диск к файловой системе, так что введите следующее:

```
mkdir /usb
mount -t vfat /dev/sdb1 /usb
cd /usb
```

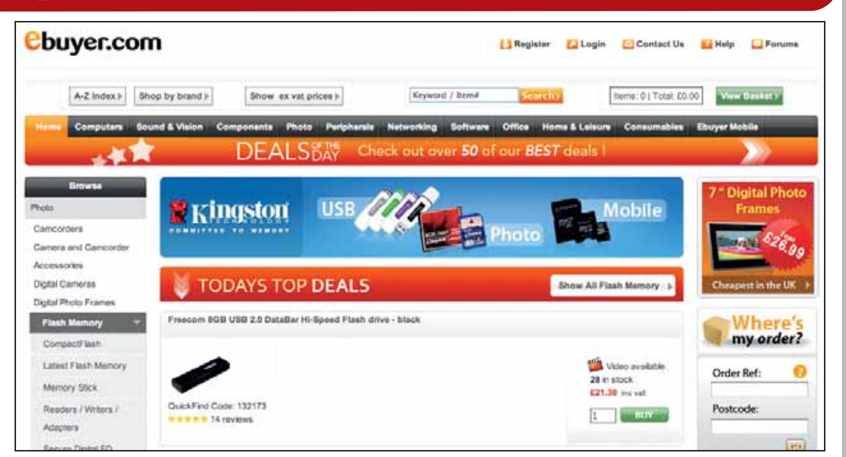
Первая команда здесь создает новый каталог **/usb**, через который мы будем получать доступ к внешнему носителю, **/usb**. Вторая команда – обычная процедура монтирования: создание привязки устройства к каталогу. Обратите внимание, что вам надо будет поменять sdb1 на

Руководство для покупателя флэш-памяти

Флэш-память, как в виде SD-карт, так и USB-брелков, сильно разнится в цене. Несколько лет назад мы все жаждали дорогих устройств на 1 ГБ. Теперь же можно выбрать 4 ГБ, казавшиеся невероятными даже два года назад, менее чем за 20 долларов. Однако есть несколько вещей, о которых следует знать. Во-первых, флэш-память имеет ограниченный срок жизни: данные на нее можно записывать конечное число раз.

Как правило, производители заявляют, что память придет в негодность после примерно 10000 записей (сохранений файлов), что кажется очень большим числом, но типовая память «no-name» сбоит намного раньше. Если вы просто хотите поиграть с другими дистрибутивами Linux, возьмите простой флэш-брелок или карту памяти. Если вам нужно сохранять важные данные, лучше подойдет широко известный брэнд, вроде Kingston. Вы заплатите немного больше (обычно – на 10 долларов для для 4 ГБ SD-карты), но оно того стоит.

Во-вторых, некоторые люди сообщали о проблемах с SD-картами на 16 ГБ и больше. Многим пользователям они служат безукоризненно, но перед тратой посетите форумы на http://forum.eeeuser.com и поищите модель, которая вас заинтересовала, или же отправьте сообщение с просьбой совета.



› www.ebuyer.com – **замечательный западный сайт для покупки карт и брелков памяти: как дешевых, ярких и без товарных знаков, так и более дорогих моделей от именитых производителей.**

sdc1 или **sdb2**, в зависимости от полученных вами ранее результатов, и что эта команда годится для разделов, отформатированных в FAT, что обычно верно для внешних USB-устройств. Если вы переформатировали ваше устройство в Linux'овскую файловую систему ext2/3, можете опустить часть команды **-t vfat**, так как она будет распознаваться автоматически.

По последней команде делается переход в каталог **/usb**. Введите **ls**, и вы увидите список файлов на вашем внешнем накопителе (или ничего, если он был только что отформатирован!). Давайте создадим новый каталог для резервной копии Eee:

```
mkdir eee_backups
cd eee_backups
```

Создание резервной копии

Устройству, представляющему внутренний флэш-накопитель Eee, в Breeezy присваивается имя **/dev/hdc**. Вы можете убедиться в этом, введя **dmesg | grep hdc** и прочитав вывод, где показана информация о нем. Обратите внимание, что Breeezy использует американскую раскладку клавиатуры, так что символ канала посреди этой команде доступен нажатием Shift+\ (вверху справа от клавиши «твердый знак»).

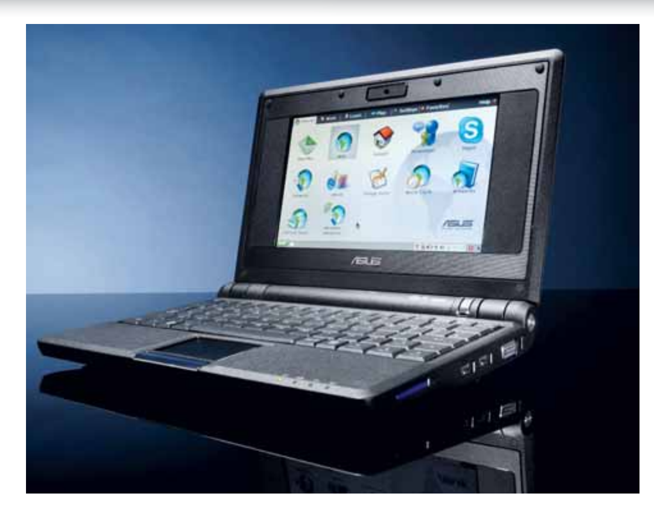
Как упоминалось выше, мы собираемся сделать точный образ внутреннего накопителя Eee. Давайте заодно сожмем его, чтобы сэкономить место и дать некоторую защиту от повреждения данных. Вот команда, которая вам нужна:

```
dd if=/dev/hdc | gzip -c5 > my_eee.gz
```

Разберемся, что здесь происходит. *dd* – это утилита Unix для копирования и преобразования файлов, и нам необходимо задать ей имя входного файла (if). Как мы видели раньше, внутренний накопитель Eee соответствует **/dev/hdc**, и мы можем использовать это имя для его чтения.

Полученные *dd* данные передаются затем посредством канала (|) в утилиту *gzip*, чтобы добавить сжатие. Ключ -c5 говорит, какой требуется уровень сжатия – его можно повысить до 9, но скорость при этом сильно снизится, так что мы выбрали среднее значение, чтобы все прошло побыстрее. Затем символ > перенаправляет вывод в файл под именем **my_eee.gz**, и, конечно, вы можете поменять его на любое другое на ваше усмотрение. Процесс копирования может занять от 10 до 20 минут, в зависимости от того, сколько данных содержится на вашем Eee.

Вот и готово: текущая установка вашего Eee теперь безопасно хранится на внешнем устройстве. Введите **cd / ; umount /usb**, чтобы отмонтировать его, после чего диск можно отключить от машины.



Скорая помощь

Окно командной строки в Breeezy может прийти в негодность и стать нечитаемым, если прокручивать много текста. Чтобы поправить это, нажмите Ctrl+L: экран очистится и ошибочные пиксели исчезнут.

Восстановление из резервной копии

Восстановление в точности противоположно Шагу 3: мы считываем сжатый файл и используем *dd*, чтобы послать вывод на внутренний накопитель Eee. Например, пусть мы хотим восстановить резервную копию с именем **xandros.gz**. Подключите ваше внешнее USB-устройство, откройте окно с командной строкой и примонтируйте его так же, как в предыдущих трех шагах. С помощью ***cd*** перейдите в каталог, где хранятся резервные копии Eee. Затем введите команду:

```
gunzip -c xandros.gz | dd of=/dev/hdc
```

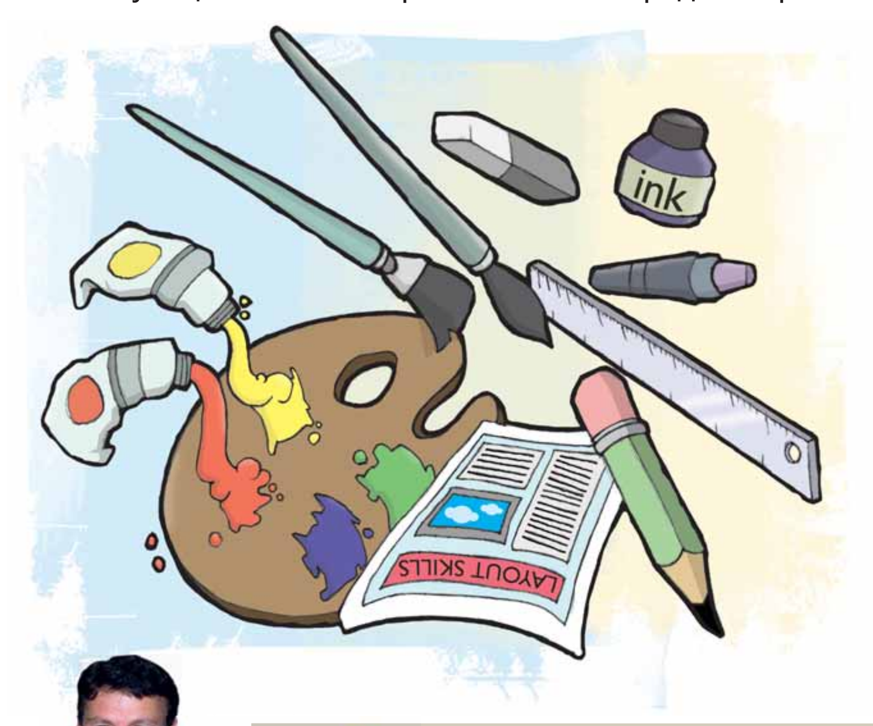
Здесь мы использовали *gunzip*, чтобы разархивировать наш сжатый файл резервной копии, затем направили его по каналу в утилиту *dd*. На этот раз *dd* требует выходной файл (of), в качестве такового мы указали внутренний накопитель Eee, **/dev/hdc**. Учтите, что для завершения этой команде может потребоваться около часа – мигание индикатора накопителя будет постоянно напоминать вам, что он занят. Теперь вы умеете создавать резервные копии и делать восстановление из них, можете устанавливать другие дистрибутивы на внутренний накопитель Eee – так что держите настрой до следующего номера, когда выйдет полное руководство! LXF

» **Через месяц** Поднимите ваш Eee на новые высоты, полностью сменив дистрибутив.

Gimp Графический редактор с открытым кодом, в который вы тоже можете вонзить свои зубы

Gimp: Работа

Как и в *Photoshop*, главное в *GIMP* – управление слоями. **Майкл Дж. Хэммел** покажет, как из манипуляций с ними вырастает ваш очередной проект.



Использование *GIMP* мало чем отличается от работы по дереву. Создавая кресло, вы определенно делаете его не из цельного бревна. Вы вырезаете и придаете форму частям, затем собираете их вместе, пока не получите окончательный продукт (тестирование на безопасность для старушек не прилагается).

Создание проекта в *GIMP* аналогично производству мебели: это тоже сборка разрозненных частей в единое целое. Сначала вы создаете части изображения вручную или берете их из хранилища фотографий, а затем совмещаете с конечным наброском в качестве слоя. Конечный набросок нуждается в небольшой правке цвета, прежде чем вы сядете на него… или, ну да, повесите его на стенку. Пускай это не совсем кресло, но идея ясна.

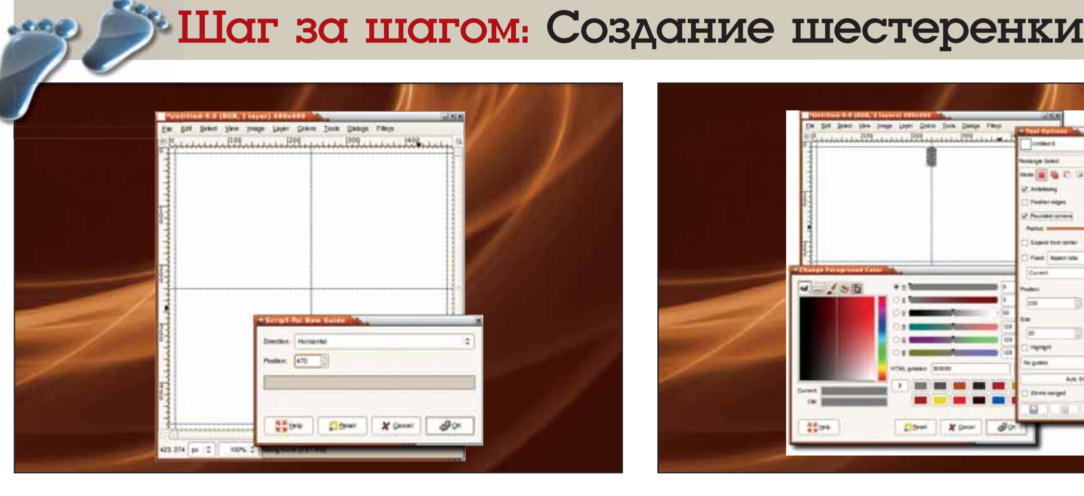
Прекрасный пример такого компонентного подхода – создание изображения работающих внутренностей механических часов. Часы состоят из шестеренок, пружин, проводов и металлического крепежа. Создание всего этого в рамках одного проекта, конечно, возможно, но это далеко не самое эффективное использование *GIMP*, вашего компьютера и особенно вашего времени. Логичнее создать шестерни, пружинки, провода и крепеж в виде независимых проектов, затем скопировать и вставить их в главный проект, где разместить как надо, подогнать размер и цвета.

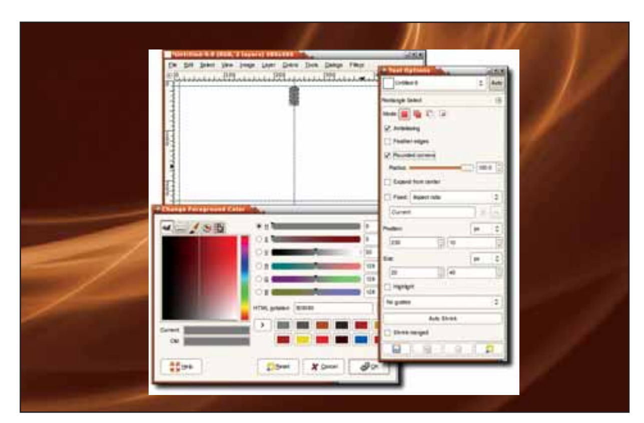
Создавая компоненты часов в качестве самостоятельных проектов,

Наш эксперт

Майкл Дж. Хэммел является участником проекта *GIMP* и автором трех книг по данной теме, включая последнюю *The Artist's Guide to Gimp Effects* [*Введение в эффекты Gimp для художников*].

Шаг за шагом: Создание шестеренки





1 Расчертите холст

Откройте диалог Создать новое изображение (Файл > Создать) и установите Ширину и Высоту по 480 точек растра (пикселей). Фон должен быть белым, чтобы легче видеть, что именно мы делаем. Нажмите кнопку OK для открытия окна нового изображения. Выберите Изображение > Направляющие > Направляющая (в %). Добавьте вертикальную и горизонтальную направляющие в позиции 50%. Затем выберите Изображение > Направляющие > Новая направляющая и добавьте горизонтальные направляющие с координатами в 10 и 470 пикселей (10 пикселей от края холста). Сделав это, добавьте вертикальные с теми же смещениями.

2 Выберите рабочую область

Создайте прозрачный слой (Слой > Создать слой). Назовите его «Зубцы». Создайте где-нибудь в изображении прямоугольное выделение. В диалоге Параметров инструмента (Диалоги > Параметры инструментов) установите Размер **20** в ширину и **40** в высоту, а Позицию – в **230** и **10**. Щелкните на опции Закругленные углы в диалоге Параметры инструмента, затем установите радиус **100.0**.

Щелкните на прямоугольнике Цвет переднего плана на панели инструментов и на вкладке HTML-нотации введите **808080**. Закройте диалог, затем перетащите прямоугольник с цветом переднего плана с панели инструментов в выделение.

Месяц назад Мы создали оттиск восковой печати, наподобие используемых для писем.

со слоями

вы также оставляете для себя возможность повторно использовать их в будущем. Сами часы, вероятно, будут иметь множество слоев, а если они предназначаются для печати, то и высокое разрешение. Открытие такого проекта может потребовать много времени и памяти. Захотите ли вы проделывать это, чтобы скопировать один слой с шестерней в другой проект? Вероятно, нет. Как я уже говорил, это неэффективное расходование ресурсов, а также неразумное применение ваших наработок. Возможность повторного использования компонентов проекта – один из способов развития вашего собственного художественного стиля.

Вернемся к часам. Шестерни – интересные компоненты данного проекта. Шестерня – это симметричный круглый объект. В квадратном окне изображения создать шестерню просто, потому что мы можем, как покажет наш урок, дублировать и вращать ее зубцы вокруг центра квадрата для размещения в нужной позиции. А предлагаемая *GIMP* гибкость в указании размера и расположения объектов делает создание шестерни еще легче.

На нашем уроке я проведу вас через создание зубцов и колец простой шестерни. Я также покажу, как залить шестерню текстурой, открыв возможность использования этого компонента в нескольких проектах.

Простенько и со вкусом

Данный проект не требует привлечения каких-либо внешних фоторесурсов или специализированных навыков в графике. Вы сможете выполнить все, будь вы гуру в *GIMP* или впервые приложили к нему руки.

Маленькие выступы по краям шестерни называются зубцами. Начнем свой проект, создав 32 зубца, равномерно распределенных по кругу. Потом соединим зубцы металлическим кольцом и закончим проект, добавив глубину, чтобы придать зубцам трехмерный вид. Работа с зубцами настолько проста, что я опишу ее еще до того, как мы начнем. Создаем один зубец вверху квадратного окна холста в собственном слое; дублируем слой и поворачиваем его на 90⁰; затем объединяем копию и оригинал. Повторим этот процесс еще раз, чтобы зубцы оказались в позициях 12, 3, 6 и 9 циферблата часов. Затем опять повторим, но уменьшим угол поворота вдвое. А на следующий раз сократим его еще в два раза. »

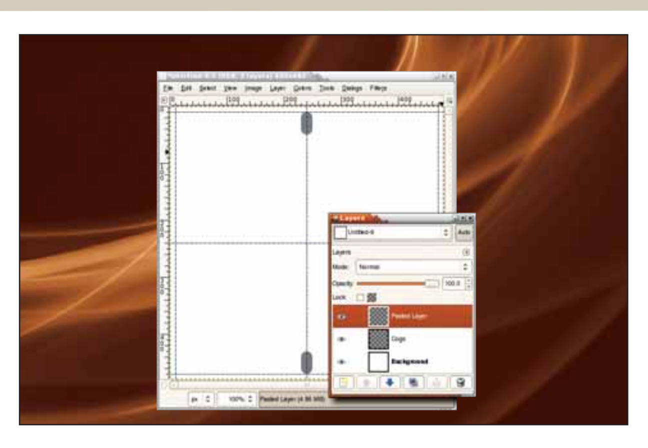
Вариации: разработка зубцов

Имеется множество вариантов, которые вы можете испробовать на этом уроке. Попытайтесь создать зубцы и разместить их ближе друг к другу. (Подсказка: просто создайте побольше зубцов). Вы также можете попробовать пилообразную форму, а любители приключений отважатся и на зубцы в форме сложных кривых. Примените творческий подход к объединению зубцов. Вместо плоского кольца или диска можно насадить зубья на витое кольцо, использовав диск и оси из данного урока.

Поищите идеи дизайна шестерни и часовых механизмов в Интернете. Кстати, и коллекции изображений содержат множество необычных форм, пригодных для замещения соединительных спиц и внутреннего диска.



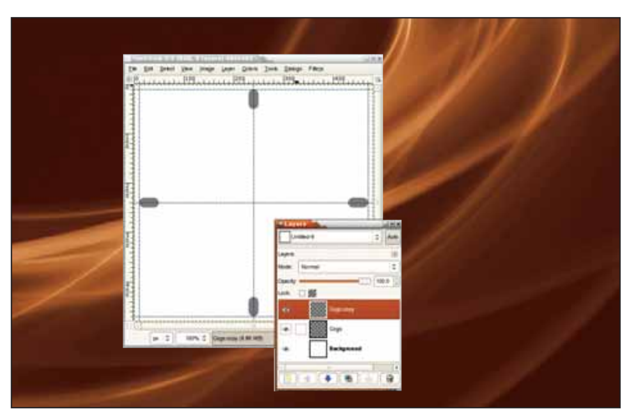
› На этом изображении много деталей, но строится оно серией простых шагов.



3 Первые два зубца

Выделение должно быть все еще активно. Скопируйте его, вставьте в окно изображения, превратите в новый слой (Слой > Создать слой) и выберите Слой > Слой к размеру изображения. Вы заметите, что маркеры границы слоя (черно-желтые штриховые линии) перескочили от краев зубца к границам холста.

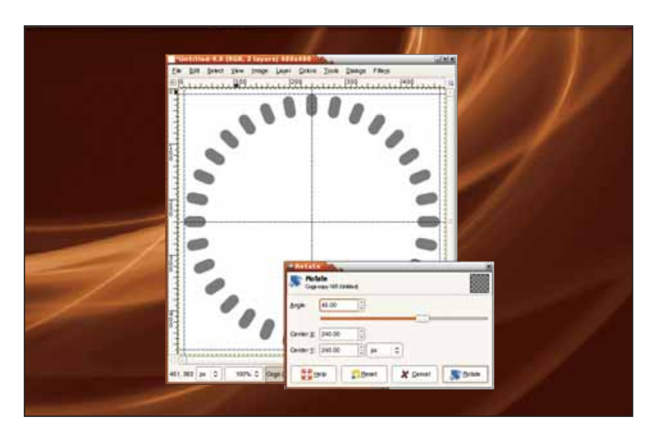
Теперь мы готовы к размещению нового зубца. Поверните слой на 180⁰ (Слой > Преобразования > Повернуть на 180), затем объедините новый слой с лежащим ниже (Слой > Объединить с предыдущим).



4 Повернуть и повторить

После первых двух зубцов остальные получить легко — путем простого копирования существующего слоя, вращения копии на соответствующий угол и слияния слоев с зубцами.

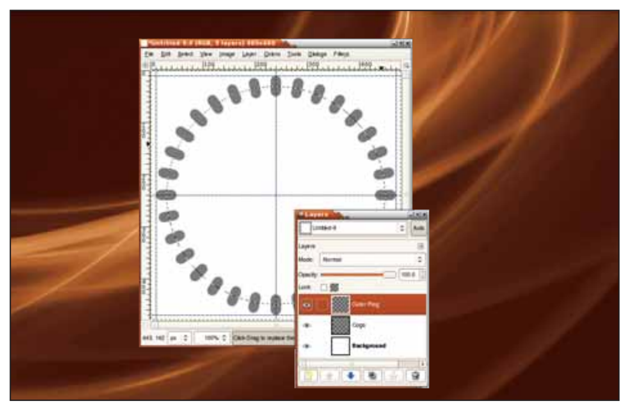
Продублируйте слой «Зубцы» (Слой > Создать копию слоя). Поверните его на 90⁰ (Слой > Преобразование > Повернуть на 90 по часовой стрелке) чтобы добавить зубцы в позиции на 3 и на 9 часов. Понятно, что на этом шаге все равно, в какую сторону вращать. Объедините новый слой с предыдущим.



5 Завершаем зубцы

Кто не уловил, последний поворот на 90⁰ есть половина первого (180⁰). Будем уменьшать поворот вдвое при каждой итерации, пока не получим все необходимые зубцы. От четырех зубцов перейдем к восьми, копируя слой, вращая его на 90/2=45 градусов и объединяя слои. Хитрость в том, что пункта меню вроде «поворот на 45» нет. Вместо этого выберите Слой > Преобразования > Произвольное вращение. В открывшемся диалоге есть поле ввода с заголовком Угол, а также ползунок. Проще набрать число (45) и нажать кнопку Повернуть.

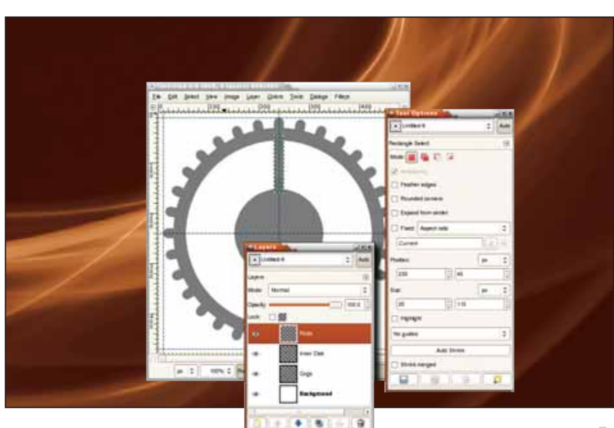
Осталось два вращения, следующие той же процедуре, что и предыдущие шаги. Для 16 зубцов: дублируем слой, вращаем на 45/2=22,5 градуса и объединяем слои. Для 32 зубцов: дублируем слой, вращаем на 22,5/2=11,25 градуса и объединяем слои. Вращения влияют на края окончательного слоя: его границы шире, чем окно изображения. Чтобы исправить это, выберите Слой > Слой к размеру изображения.



6 Свяжите их...

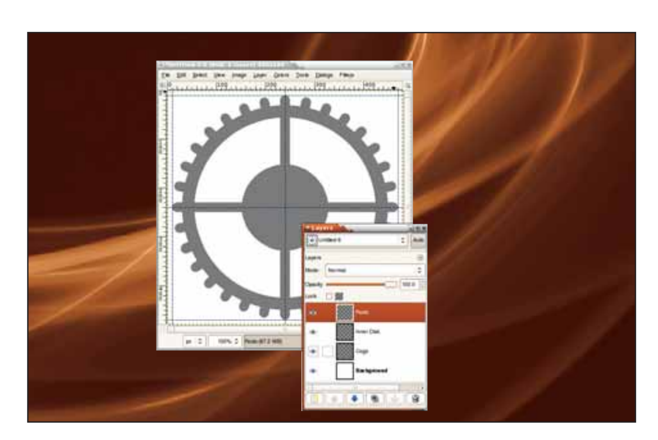
Следующий шаг в разработке шестерни – объединение всех зубьев. Мы сделаем это, создав сплошной диск, затем вырежем из него внутренний диск для создания кольца в собственном слое. Затем мы сможем объединить слои кольца и зубцов. Начните с добавления прозрачного слоя с именем «Внешнее кольцо».

Выберите инструмент Эллиптическое выделение на панели инструментов. Наши направляющие все еще в окне изображения; используем их для создания круга, выровненного по зубцам. В окне изображения щелкните на левом верхнем пересечении верхней и левой направляющих, затем растяните до пересечения с правой и нижней направляющей. Полученный круг полностью закрывает зубцы, так что изменим его размер. Высота зубцов 40 пикселей, и уменьшить круг надо на половину этой величины. Выберите Выделение > Уменьшить и введите размер **20**. Затем нажмите кнопку OK для уменьшения выделения.



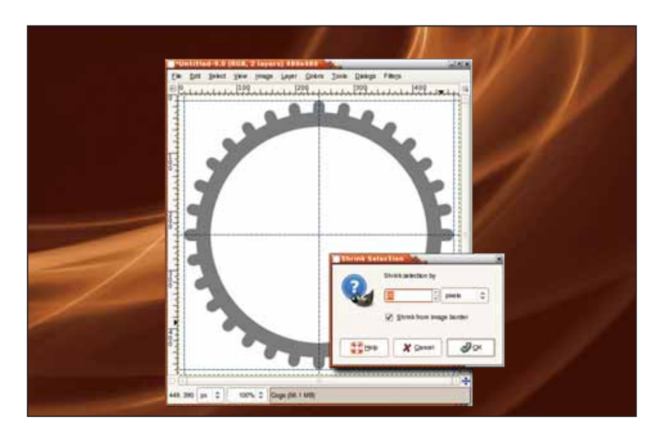
9 Спицы

До присоединения внутреннего диска, добавим несколько соединительных спиц между внутренним диском и внешним кольцом. Можно использовать те же подходы, что и при создании начального зубца: определение размера и позиции выделения вручную для создания первой спицы, затем дублирование и вращение каждого продублированного слоя. Выберите прямоугольное выделение на панели инструментов. В диалоге Параметры инструмента (Диалоги > Параметры инструментов) установите Размер **20** в ширину и **115** в высоту и Позицию **230** и **45**. После этого выделение будет размещено по центру вертикальной направляющей между внешним кольцом и внутренним диском. Пока отключим опцию Закругленные углы (щелкните на ней, чтобы убрать флажок). Добавьте прозрачный слой, назовите «Спицы», затем залейте выделение цветом переднего плана.



10 Все оформлено

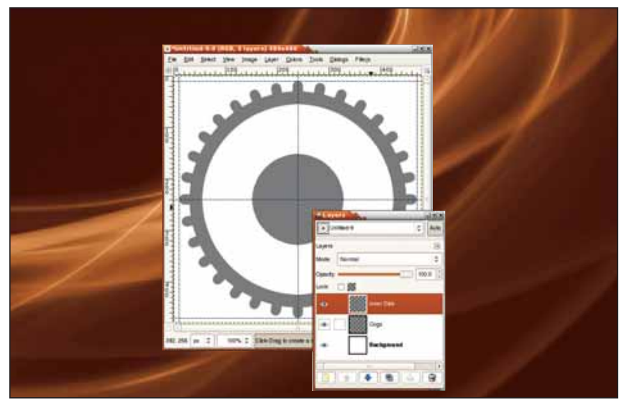
Выделение все еще активно, так что скопируйте и вставьте его как начальный зубец. Преобразуйте плавающее выделение (оно будет создано при вставке) в новый слой. Установите границы слоя по размеру изображения (Слой > Слой к размеру изображения). Теперь переверните вставленный слой в вертикальной плоскости (Слой > Преобразования > Отразить по вертикали). Объедините два слоя спиц. Вновь продублируйте объединенный слой. Поверните его на 90⁰. Опять объедините два слоя спиц. Наконец, объедините слой «Спицы» со слоем «Внутренний диск», а слой «Внутренний диск» – со слоем «Зубцы». Вот шаблон для нашей шестерни. Теперь мы можем придать ей объем и текстуру.



7 Вырежем диск

Цвет переднего плана должен был остаться тем же, что мы взяли для зубцов, так что перенесите его прямо в выделение. Теперь нужно вырезать из этого выделения диск. Помните, что высота зубцов 40 пикселей, так что ширина внешнего кольца должна быть по крайней мере 20. Возьмем чуть больше: уменьшим выделение на 25 пикселей. Для завершения внешнего кольца осталось вырезать выделение (Правка > Вырезать или Ctrl+X) из существующего серого диска. Теперь объедините слой зубцов со слоем внешнего кольца.

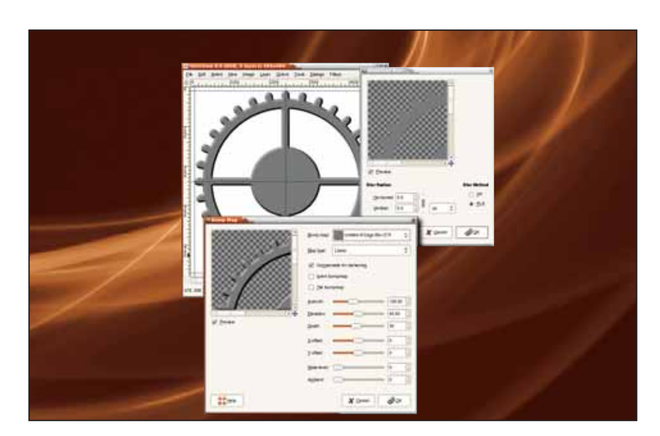
Затем добавим к шестерне внутреннее кольцо. Учтите, что все, кроме зубцов, формы шестерни (внутренние кольца и украшения) – целиком выбор художника. Вы можете, чтобы поупражнять мастерство, попытаться скопировать настоящую шестерню вплоть до мельчайших деталей или создать внутри внешнего кольца формы, не являющиеся однородными дисками и спицами. Однако мы для вашей первой шестерни оставим все простым и применим формы, которые легко создавать и дублировать.



8 Немного математики!

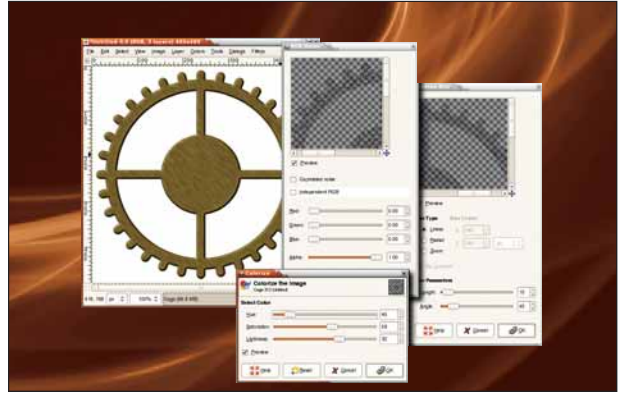
Внутренний диск нашей шестерни начинается с другого овального выделения. Наилегчайший способ создать его – выделить прозрачную область внутри внешнего кольца слоя зубцов. Данный слой активен, так что вы можете выбрать на панели инструментов Выделение связанной области. Щелкните в окне изображения внутри внешнего кольца. Это создаст выделение, следующее по внутреннему краю кольца. Добавьте прозрачный слой, где мы создадим внутренний диск, и назовите его «Внутренний диск». Однако нам надо, чтобы внутренний диск был значительно меньше, чем это выделение. Пора заняться математикой.

Вспомните, что окно изображения – квадрат со стороной 480 пикселей. Кроме того, мы создали зубец с высотой 40 пикселей и разместили его на 10 пикселей ниже (по вертикали) в окне изображения. Это означает, что нижний край зубца на 50 пикселей ниже верхнего края изображения. Центр изображения по вертикали располагается в 480/2=240-м пикселе. То есть текущее выделение имеет радиус 240-50 = 190 пикселей. Уменьшим его вдвое до 95 пикселей, то есть мы хотим уменьшить выделение на 95 пикселей и залить его серым. Очистите выделение (Выделение > Снять или Ctrl+Shift+A).



11 Создаем объем…

Продублируйте слой «Зубцы» и назовите его «Размытые зубцы». Примените Гауссово размытие (Фильтры > Размытие > Гауссово размытие) на пять пикселей по вертикали и горизонтали. Отключите видимость размытого слоя в диалоге Слои. Выберите слой Зубцы. Откройте фильтр Карта рельефа (Фильтры > Карта > Карта Рельефа). Введите параметры рельефа для слоя «Размытые зубцы». Установите значения Азимут **135**, Возвышение **45** и Глубина **30**. Нажмите OK для применения этих настроек – слой «Зубцы» примет объемный вид.



12 …и текстуру

Теперь добавьте небольшую текстуру. Для выбора всей шестерни, щелкните на панели инструмент Выделение связанной области и затем, удерживая клавишу Shift, кликните на всех белых областях изображения. Инвертируйте выделение (Выделение > Инвертировать). Теперь шестерня выделена. Создайте прозрачный слой и назовите его «Шум». Используйте фильтр Шум RGB (Фильтры > Шум > Шум RGB) для заполнения выделения шумом. Отключите опцию Независимый шум и сдвиньте все ползунки к левому краю (установите значения **0**) за исключением ползунка Альфа, который должен быть у правого края.

Добавьте размывание движением (Фильтры > Размывание > Размывание движением) к слою «Шум». Установите длину **10** и угол **45**. Примените это к слою «Шум». Объедините слои «Шум» и «Зубцы». Откройте диалог Тонировать и выберите подходящий для шестерни цвет. Снимите выделение (Выделение > Снять), затем откиньтесь на спинку кресла и восторгайтесь своим творением. LXF

Через месяц Владельцы цифровых камер – перенесите свои фотографии прямо в *GIMP*.

Rails: Создаем

Страшитесь С? Путаетесь в Perl? Пусть **Алекс Янг** покажет вам, как просто и быстро начать программировать на Ruby.



С момента своего создания фотогалерея Flickr поразительно разрослась. Она полностью изменила наши представления о хранении снимков в Интернете, а через ее каталоги каждый посетитель мог достаточно просто найти самые интересные фотографии. Но как ни хороша Flickr, разве не здорово иметь на своем сайте личную фотогалерею? Можно реализовать те возможности, которые нужны именно вам, и даже поделиться кодом с друзьями, чтобы они тоже создали себе нечто подобное.

Для написания классной, продвинутой галереи мы воспользуемся Ruby on Rails. Rails избавляет вас от мучений, присущих обычной web-разработке – можете забыть о мороке с SQL и XML; Rails предоставляет простые и удобные для восприятия абстракции через дружелюбный к программисту код Ruby. Его легко установить – потребуются лишь база данных и web-сервер; и web-разработка упрощается, благодаря наличию готовой структуры для каждого приложения.

Rails испытывает влияние популярных методологий: Не Повторяй Самого Себя (Don't Repeat Yourself – DRY) и Разработка Через Тестирование (Test Driven Development – TDD). DRY предполагает, что код не должен дублироваться, и это уменьшает общую сложность проекта. TDD часто практикуется профессиональными Rails-разработчиками: данный метод подразумевает тестирование кода до его завершения. Написание тестов до того, как написано само приложение, часто способствует разработке более API-подобного кода с более ясной инкапсуляцией.

Мы рассмотрим некоторые технологии, используемые Rails-разработчиками в реальных проектах:

» Установим Ruby on Rails с библиотеками и модулями расширения для обработки изображений.
» Создадим фотогалерею с миниатюрами и возможностью управления пользователями.
» Ознакомимся с основными компонентами каркаса Rails.

Наш эксперт

Алекс Янг – технический писатель и ведущий инженер Helicoid Limited, компании, специализирующейся на разработке web-приложений на Ruby on Rails. Алекс регулярно пишет статьи о Rails для сайта http://alexyoung.org.

Часть 1 Устанавливаем Rails

Первая установка Rails кажется сложнее, чем она есть на самом деле – ведь познакомиться придется всего лишь с несколькими технологиями (если вы уже не программист Ruby!). Чтобы установить Rails, отвечающий потребностям нашего проекта, потребуются пять компонентов:

» Ruby.
» *RubyGems*.
» Ruby on Rails: http://rubyonrails.org.
» Реляционная СУБД – *MySQL*, *PostgreSQL*, *SQLite* (сегодня мы будем пользоваться *SQLite*).
» Прочие библиотеки для обработки изображений и работы с Rails.

Минут через 15 все должно быть готово к написанию кода.

Для начала нужно проверить, установлен ли Ruby: команда **which ruby** выведет путь к интерпретатору. Если его у вас нет, понадобится пакет Ruby, и он есть в большинстве дистрибутивов Linux. В Debian и Ubuntu Ruby можно установить с помощью команды

```
sudo apt-get install ruby1.8
```

Желаете собрать Ruby из исходников? Загрузите их с сайта www.rubylang.org/en. Затем вам понадобится *RubyGems* – менеджер пакетов, с помощью которого можно устанавливать, удалять и обновлять библиотеки Ruby. В большинстве дистрибутивов Linux есть пакет для *RubyGems*: например, в Debian и Ubuntu это **libgems-ruby1.8**. Однако на момент написания статьи пакеты в этих дистрибутивах сильно устарели и не будут корректно работать с Rails 2, поэтому лучше установите Gems из исходных текстов.

Загрузите архив с сайта www.rubygems.org и распакуйте его, а затем запустите из получившегося каталога следующую команду:

```
sudo ruby setup.rb
```

Она установит команду *gem1.8* и связанные с ней библиотеки.

После установки *RubyGems* добавление Rails – сущий пустяк. Следующая команда инсталлирует все необходимые библиотеки Ruby, а также утилиты командной строки Rails:

```
sudo gem1.8 install rails rake
```

фотоальбом



Для хранения данных моделей Rails также необходима реляционная БД. Если вам кажется, что сервер СУБД на рабочем столе – излишество, есть прекрасная альтернатива для локальной разработки – *SQLite* (www.sqlite.org). Удостоверьтесь, что установлены как *SQLite*, так и ее библиотеки для разработчиков. Для этого выполните команды:

```
sudo apt-get install sqlite3
sudo apt-get install libsqlite3-dev
sudo gem1.8 install sqlite3-ruby
```

Убедимся, что все работает

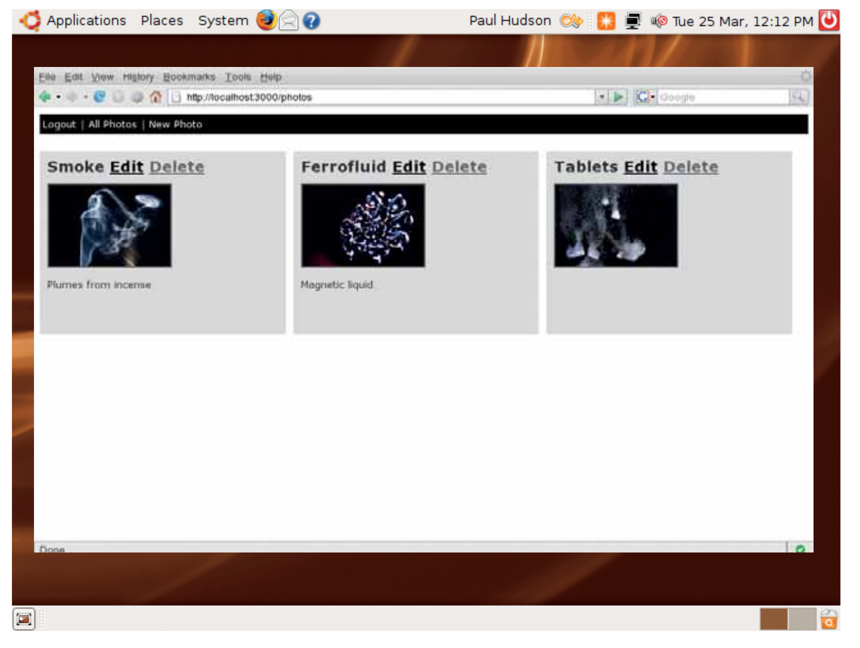
Теперь давайте проверим, что Rails установлен корректно. Чтобы начать проект Rails, просто наберите **rails gallery** – при этом будут созданы все необходимые каталоги и файлы конфигурации:

```
rails gallery
cd gallery
script/server
```

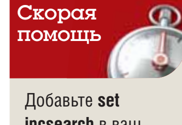
Последняя строка запускает web-сервер на Ruby, его можно использовать при разработке проекта. Откройте запущенный проект в браузере (http://localhost:3000). Вы должны увидеть страницу приглашения Rails. Пусть она висит в фоне, пока вы заняты этим уроком. Согласно настройкам БД по умолчанию, в подкаталоге **db/** каталога проекта будет автоматически создана база данных *SQLite3*. Если вы хотите использовать вместо нее *MySQL* или *Postgres*, отредактируйте файл **config/database.yml**, указав в нем настройки своей БД. Убедитесь, что переменная adapter установлена в **mysql** или **postgresql**.

Наконец, фотогалерея будет бесполезной, если в ней не будет базовых возможностей обработки изображений. Для создания высококачественных миниатюр, воспользуемся библиотеками Rails *ImageMagick* и *RMagick*:

```
sudo apt-get install libmagick9-dev
sudo gem install rmagick
```



› Завершенная галерея – с загрузкой файлов, миниатюрами и навигацией.



Скорая помощь

Добавьте **set incsearch** в ваш vimrc для поиска по мере ввода текста и используйте Ctrl+P для подстановки.

Часть 2 Приступаем к делу

Теперь пора показать, какая это стоящая и простая штука – разработка с Rails. Rails содержит средства генерации скелетных файлов, позволяющих создавать полнофункциональные приложения, а также поддерживает модули расширения. Мы воспользуемся и тем, и другим, чтобы написать галерею быстрее, чем разработчик на C# загрузит *Visual Studio*.

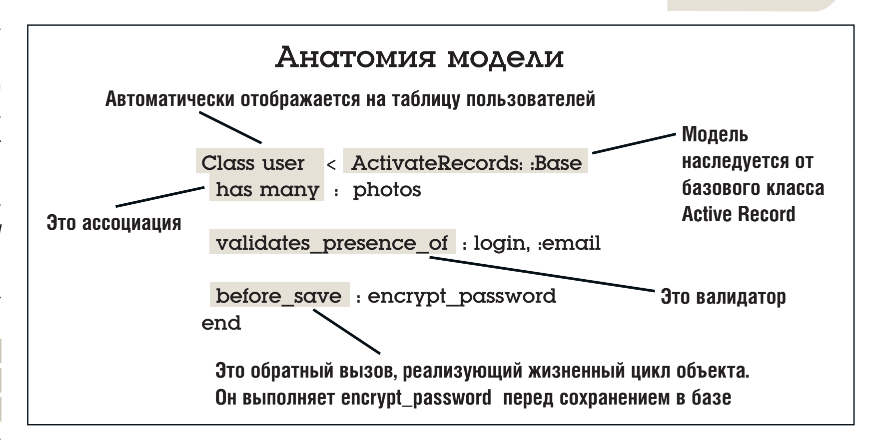
Если бы я сел и написал набор утилит для управления пользователями, у меня бы вышло нечто очень похожее на модуль *Restful Authentication* Рика Олсона [Rick Olson] – ну так сэкономим время и усилия и сразу возьмем его. Из подкаталога **gallery/** каталога проекта выполните команды:

```
script/plugin source http://svn.techno-weenie.net/projects/plugins
script/plugin install restful_authentication
script/generate authenticated user sessions
```

Модуль *Restful Authentication* предоставляет: модель БД для пользователя; миграции БД для создания пользователя; контроллеры пользователя и сессии для обработки создания учетной записи, вход в систему и запоминание пользователя в cookies.

Модули устанавливаются в подкаталог **vendor/plugins/** каталога проекта. Данный конкретный экземпляр также добавляет файлы в код проекта в каталоге **app/** с помощью генератора.

Теперь откроем файл **app/models/user.rb** и рассмотрим код модуля. Он представляет несколько общеупотребимых технологий для управления объектами ActiveRecord. ActiveRecord – одна из основных частей каркаса: она отображает объекты Ruby на базу данных.



Даже с небольшим опытом в Ruby или его полным отсутствием вы должны понять следующий код:

```
validates_presence_of :login, :email
before_save :encrypt_password
```

Эти строки – на самом деле методы, которые выглядят как макросы; это общепринятый стиль в Rails. С помощью методов validates_ »

Анатомия пути

map. edit_photo 'photos/:id/edit' , : controller => 'photos', : actions => 'edit'

- **edit_photo** — **Это название пути. Вызов edit_url{@photo} в любом месте приложения выведет путь**
- **'photos/:id/edit'** — **Это сам путь. Параметры отмечаются двоеточием**
- **controller** — **Это название контроллера**
- **actions** — **Это название действия (по умолчанию – "index")**

» легко проверить поля перед их сохранением в базе данных. Метод before_save реализует жизненный цикл объекта – он позволяет вам выполнять свой собственный код при сохранении данных в базе. Здесь перед записью модели в БД будет вызван метод encrypt_password.

Прежде чем продолжить, запустите процедуру миграции, полученную от модуля:

```
rake db:migrate
```

Она выведет некоторые данные о таблице:

```
1 CreateUsers: migrating
-- create_table("users", {:force=>true})
-> 0.0044s
1 CreateUsers: migrated (0.0046s)
```

Добавим пользователя и зайдем

Но как нам воспользоваться тем, что мы успели добавить? К счастью, модуль *Restful Authentication* установил контроллер, поддерживающий создание пользователя. Контроллеры – другая важная часть Rails: здесь они известны как ActionControllers и находятся между HTML-шаблонами и моделями ActiveRecord.

Чтобы добавить пользователя, зайдите на http://localhost:3000/users/new в своем браузере и введите необходимые данные. Затем укажите выбранное вами имя на http://localhost:3000/sessions/new. Вы будете отосланы на страницу шаблона Rails, но после добавления фотографий мы это исправим. Выполните команду:

```
script/generate scaffold photo title:string description:text filename:
string content_type:string size:integer user_id:integer width:
integer height:integer parent_id:integer path:string thumbnail:string
rake db:migrate
```

Будут добавлены: модель Photo; таблица в БД для фотографий с ID (создается автоматически) и набором необходимых полей; контроллер PhotosController с действиями по просмотру, созданию, обновлению и удалению фотографий; и сопутствующие представления (виды). Чтобы взглянуть на результаты, откройте адрес http://localhost:3000/photos. Хотя это и работает, но выглядит не очень. Опять же и контроллер пользователя не сочетается с сайтом. Давайте решим эти проблемы…

Попробуйте Ruby



Вы новичок в Ruby и хотите попробовать язык, ничего не устанавливая? Try Ruby (http://tryruby.hobix.com) – это web-консоль Ruby, с помощью которой можно попробовать язык в браузере.

Добавляем стиль

Генератор каркаса создал файл **app/views/layouts/photos.html.erb**. В директории раскладок сайта находятся различные HTML-шаблоны. Используйте только один для вашего приложения, и он будет разделен между контроллерами.

Все контроллеры по умолчанию используют шаблон **application.html.erb**, если не задано иное, поэтому переименуйте **photos.html.erb** в **application.html.erb**, откройте его и измените содержимое тэга <title> на что-нибудь более общее. Открыв http://localhost:3000/users в своем браузере, вы увидите пользовательские шаблоны с **application.html.erb**.

Теперь нужно лишь задать индексную страницу сайта. Используем в качестве нее список фотографий. Удалите файл **public/index.html**, который был установлен Rails, и добавьте следующую строку в файл **config/routes.rb**:

```
map.connect '', :controller => 'photos'
```

Rails предоставляет все, что нужно для обработки загрузки файлов. Однако вместо реализации ее в PhotosController (сокращение контроллера и повторное использование кода в моделях – хорошая практика), можно поместить большую часть в модель Photo. Так как все загрузки файлов обрабатываются в общем-то одинаково, для добавления в модель необходимого функционала можно воспользоваться модулем расширения. Установите модуль *attachment_fu*:

```
script/plugin install attachment_fu
```

Теперь откройте модель Post (**app/models/post.rb**) и настройте его:

```
class Photo < ActiveRecord::Base
 has_attachment :storage => :file_system,
  :thumbnails => { :thumb => '160>' },
  :content_type => :image
 validates_as_attachment
end
```

Открыв модель Photo, добавьте следующую строку:

```
belongs_to :user
```

Будет установлена связь между фотографиями и пользователями. Также можно добавить has_many :photos в модель User.

Просмотр фотографий

Перед тестированием нужно исправить виды 'photo edit' и 'new', активировав в них загрузку файлов. Откройте **views/photos/edit.html.erb** и **views/photos/new.html.erb** и измените определение формы на следующее (это просто стенографическая запись Rails для HTML-кода):

```
<% form_for(@photo, :html => { :multipart => true }) do |f| %>
```

Некоторые из полей, созданных генератором, не подходят для наших форм. Уберите все поля после Description в обоих шаблонах – просто удалите целый блок для каждого поля, включая тэг параграфа. Наконец, добавьте поле загрузки файла:

```
<p>
<b>Photo</b><br />
<%= f.file_field :uploaded_data %>
</p>
```

Теперь нужно изменить индексную страницу, содержащую список фотографий – так, чтобы на ней показывались миниатюры. Замените весь файл строкой:

```
render :partial => 'photo', :collection => @photos %>
```

Эта строка выводит 'partial' (так называются повторно используемые шаблоны) для каждой фотографии в массиве @photos. Контроллер PhotosController загружает @photos в действии для индексной страницы. Добавить partial для фотографий просто: создайте файл **app/views/photos/_photo.html.erb** и добавьте в него строки:

```
<div class="photo">
<h2>
<%= h photo.title %> <%= link_to 'Edit', edit_photo_path(photo) %>
<%= link_to 'Delete', photo, :confirm => 'Are you sure?', :method =>
:delete %>
</h2>
<%= link_to image_tag(photo.public_filename(:thumb)), photo %>
```

```
<%= textilize h(truncate(photo.description)) %>
</div>
```

Некоторые из вызовов этих методов – помощники, предоставляемые Rails и библиотеками, которые он использует. Например, h() гарантирует, что текст безопасен и не содержит HTML-кода – это хороший способ «провести санобработку» отправляемого пользователем контента. Метод textilize выводит текст в соответствии с требованиями Textile.

Также можно отредактировать файл **app/views/photos/show.html.erb**, удалив ненужные поля. Я убрал заголовок ('title'), переместил ссылки «редактировать/назад» в верхнюю часть файла и добавил тэг image для фотографии:

```
<h2><%=h @photo.title %></h2>
<%= link_to 'Edit', edit_photo_path(@photo) %> |
<%= link_to 'Back', photos_path %>
<div class="large_photo"><%= image_tag @photo.public_filename %></
div>
```

Теперь измените основной шаблон (**app/views/layouts/application.html.erb**), исправив строку stylesheet_link_tag на…

```
<%= stylesheet_link_tag :all %>
```

…и добавив некую общую навигацию над выражением yield:

```
<div id="navigation">
<% if current_user %>
<%= link_to 'Logout', session_url, :method => 'delete' %> |
<% else %>
<%= link_to 'Login', new_session_url %> | <%= link_to 'Register', new_
user_url %> |
<% end %>

<%= link_to 'All Photos', photos_url %> | <%= link_to 'New Photo', new_
photo_path %>
</div>
```

Вы будете часто использовать link_to. Он создает ссылки на основе файла **config/routes.rb**, гарантируя, что URL-адреса будут выглядеть именно так, как вы хотите, и позволяя вам забыть о том, как они пишутся. Если вы видите метод, который заканчивается на _url или _path, знайте, что это ссылка на путь. Например, photos_url и photos_path оба вернут **/photos**.

Контроллер ApplicationController определяется в файле **app/controllers/application.rb** и наследует от ActionController::Base. Это действие, которое по умолчанию визуализирует шаблон **app/views/photos/index.erb.html**. PhotosController доступен по адресу http://localhost:3000/photos. Чтобы миниатюры не выводились на индексной странице, контроллер нужно изменить. Отредактируйте файл **app/controllers/photos_controller.rb** так, чтобы вывод индекса игнорировал миниатюры, где-то в строке 5:

```
@photos = Photo.find(:all, :conditions => 'thumbnail is NULL')
```

Анатомия контоллера

PhotosController доступен по адресу http://localhost:3000/photos.

```
class PhotosController < ApplicationController
  before_filter : login_required

  def index
  end
end
```

Это действие. По умолчанию оно визуализирует шаблон app/views/photos/index.erb.html.

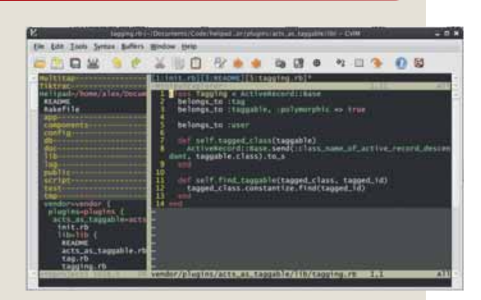
app/controllers/application.rb определяет ApplicationConroller, который наследует от ActionController::Base

Искусное редактирование текста в Linux

Большинство обучающих роликов по Rails записаны в *TextMate* и Mac OS X. Пользователи Linux могут сделать примерно то же самое (и, возможно, пойти дальше), используя *Vim* с этими замечательными модулями расширения и возможностями самого *Vim*:

» **Project** Быстро систематизирует файлы проекта Rails и производит поиск в них: http://vim.sourceforge.net/scripts/script.php?script_id=69.

» **PMiniBufExplorer** Добавляет вкладки для быстрого переключения между открытыми файлами: www.vim.org/scripts/script.php?script_id=159.



› **Организовать свои файлы Rails будет проще с помощью модулей Vim.**

Перед отправкой фотографий также нужно убедиться в том, что пользователь зашел в систему. Удалите следующие строки из UsersController и добавьте их в **app/controllers/application.rb**:

```
# Be sure to include AuthenticationSystem in Application
# Controller instead
include AuthenticatedSystem
```

Теперь нужно удостовериться, что при создании пользователя устанавливается его ID. Замените первую строку действия create в PhotosController:

```
@photo = current_user.photos.new params[:photo]
```

и измените первую строку действия update:

```
@photo = current_user.photos.find params[:id]
```

Как видите, здесь, для гарантии того, что пользователь может редактировать только свои фотографии, используется связь Photos с моделью User. Нужно сделать еще кое-что: убедиться, что только пользователи, зашедшие в систему, могут создавать и редактировать фотографии. Это можно сделать с помощью модуля *Restful Authentication*. Добавьте такую строку над выводом индекса в PhotosController:

```
before_filter :login_required, :only => [ :edit, :update, :new, :create ]
```

Стили фотографий

Таблицы стилей хранятся в каталоге **public/stylesheets**. Добавьте туда файл **screen.css**:

```
#navigation { background-color: #000; padding: 4px; margin: 0; color:
#aaa; }
#navigation a { color: #fff; text-decoration: none; }
#navigation a:hover { background-color: #fff; color: #000; }
.photo { width: 30%; float: left; background-color: #f0f0f0; margin:
10px 10px 0 0; padding: 10px; height: 220px; }
.photo h2 { margin: 0 0 14px 0; padding: 0; font-size: 1.5em; }
.large_photo { margin: 10px 0; }
```

Остановите **script/server** на своем терминале, нажав Ctrl+C. Теперь снова запустите его, и *attachment-fu* будет готов к использованию в вашей фотогалерее.

После добавления модулей в проект нужно перегрузить процесс web-сервера; то же относится к изменению файлов конфигурации Rails, включая **config/environment.rb**. К счастью, остальной код и **config/routes.rb** редактируются динамически.

Теперь можно кликнуть по ссылке Register для создания пользователя и зайти в систему. Помните, что при этом используется имя, а не адрес электронной почты. Кликните по ссылке New Photo для загрузки фотографии в коллекцию. Теперь используйте сэкономленное время: загрузите свои отпускные снимки! LXF



Vim 7 поддерживает вкладки **c :h tabnew**.

» **Через месяц** Мы добавим в наш проект немного Web 2.0

Говорящие

Всемирная сеть может говорить с тобой! **Грэм Моррисон** озвучит новостную ленту RSS, используя простой скрипт Python и речевой синтезатор *Festival*.



На первый взгляд, написание скрипта/программы для скачивания и разбора RSS-ленты и последующая отправка новостей в синтезатор речи – задача амбициозная, даже для наших продвинутых уроков. Но, как выясняется, на самом деле это довольно просто, в основном благодаря трем технологиям: Python, *Festival* и Linux. Python – доминирующий в мире скриптовый язык – позволяет легко создавать сценарии, особо не задумываясь и не прилагая чрезмерных усилий. Система синтеза речи *Festival Speech Synthesis System* с открытым исходным кодом – фантастическая штука, и ее можно установить парой щелчков мыши из менеджера пакетов вашего дистрибутива. Да и сам Linux – без его мощных каналов и планирования процессов у нас ушло бы куда больше времени на реализацию всего этого функционала в нашей программе, и мы также должны были бы сами добавить к ней графический интерфейс, чтобы сделать ее легко доступной. К счастью, все, что надо сделать – это накропать небольшой скрипт на Python и немного поколдовать с командной строкой для увязывания всего этого вместе. Мы напишем сценарий Python, выводящий сообщение в текстовом формате, а затем подадим текст на вход *Festival*, который озвучит новость через динамики или наушники. Это дает максимальную гибкость. Такой двуединый подход (Python-скрипт, передающий вывод в *Festival*) можно применять к практически любой цели. Менее чем через час вы сможете сидеть сложа руки и слушать нежный женский голос синтезатора, читающего последние события с вашего любимого сайта.



Наш эксперт

Грэм Моррисон Разработчик свободного ПО, фанат KDE и энтузиаст электронной музыки. Он обсуждал синтезаторы не с одним ведущим членом сообщества FOSS.

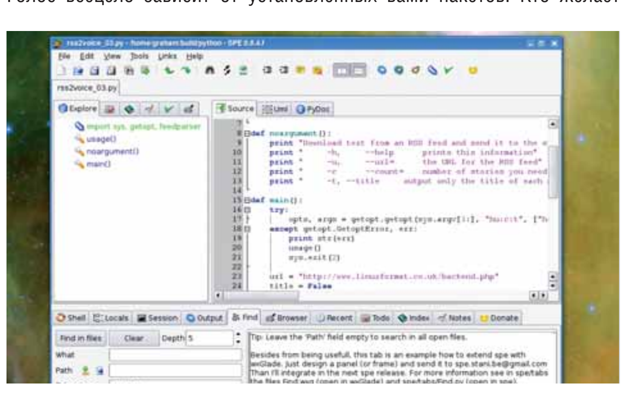
Что вам нужно:

- Python 2.5.2
- *Feedparser* for Python
- *Festival 1.96 beta*
- *aplay* (from *alsa-utils*)

Часть 1 Говорящие головы

Для начала нужно заставить синтезатор речи работать. Может показаться, что мы начинаем с конца, но ведь без настроенного синтезатора речи мы не сможем построить и опробовать нашу программу. Пакеты *Festival* включены в большинство популярных дистрибутивов Linux, и это прекрасно, по сравнению с запутанными требованиями и номенклатурой, используемыми на web-сайте проекта. *Festival* может «говорить» на различных языках и диалектах, от хинди и маратхи до чешского и итальянского; это означает, что нужный для установки пакет будет зависеть от вашей локали [русский также доступен, – *прим.ред.*]. Для классического английского [UK-English], например, вам нужно установить два пакета: '*Festival* lexicon from the Oxford Advanced Learners' Dictionary' и 'Part of speech lexicons and ngram from English', а также основной пакет *Festival* и пакет «диктора» (голос) по вашему выбору. В зависимости от популярности вашего языка, вы можете найти мужские и женские голоса, с низкой (8000) и высокой (16000) частотой дискретизации.

Тем, кто никогда раньше не пользовался *Festival*, он покажется немного загадочным. Заставить программу заговорить, похоже, было для разработчиков побочной целью. Наберите **festival** в командной строке, и вы увидите приглашение интерфейса интерпретатора. Вместо «здрасьте», напечатайте **(SayText "Hello World!")**, включая скобки. Если все установлено и работает правильно, вы услышите утробный голос синтезатора речи *Festival*, изрекающий слова «hello» и «world». Голос всецело зависит от установленных вами пакетов. Кто желает



› Надоело набирать скрипты Python? Используйте IDE Python, например, *SPE*. Графический интерфейс для вашего скрипта помогут создать eric для разработки на *Qt* или PIDA для *GTK+*.

» **Месяц назад** Под капотом: мы разбирались с модулями и ядрами.

Новости

ужаснуться количеству предлагаемых *Festival* вариантов, просто наберите **help** для их просмотра. Нам же нужно, чтобы *Festival* вещал из командной строки, и для этого есть два разных метода. Первый использует аргумент **--tts**. Эта команда будет читать либо файл (дословно!), либо стандартный вывод с терминала. То есть мы можем канализовать выход любой команды в *Festival*. Например, набрав **echo "Hello world" | festival --tts**, вы получите результат, аналогичный предыдущему: здесь мы перенаправляем вывод команды **echo** в *Festival* без запуска интерпретатора, а он затем передает звук в ваши динамики. Но у этой тактики есть проблемы. Если звуковая карта уже используется, то вы получите ошибку: "Linux: can't open /dev/dsp", потому что *Festival* не взаимодействует со звуковой подсистемой Linux – он просто посылает необработанные аудиоданные на **/dev/dsp**, а устройство занято, и это не работает. Большинство окружений рабочего стола любят захватывать **/dev/dsp**, так что при получении нового сообщения вместо новости вы услышите разве что звук «пинг».

Решение немного запутанно. Во-первых, запустим *Festival* как сервер, набрав **festival --server**. Это заставит приложение работать в фоновом режиме, молча ожидая от клиента *Festival* просьбы перевести текст в речь. Такая схема более эффективна, чем запуск *Festival* при каждой необходимости что-то озвучить, и это также открывает возможности для дистанционных монологов. Для подключения к серверу и генерации речи используем команду **festival_client** в другом терминале, а также аргумент **--ttw**. Вместо того, чтобы отправлять звук прямо на аудиоустройство (и сбоить, если устройство уже занято), опция **ttw** будет передавать необработанные аудиоданные на консоль. Нам надо перенаправить их в команду, способную прочитать этот вывод и отправить его в аудиоподсистему. Простейший вариант – команда **aplay**; скорее всего, она установлена по умолчанию. *Aplay* общается с ALSA напрямую, а значит, вы можете поделить свое аудиоустройство между приложениями. Полная версия команды, использующая клиент и сервер и посылающая вывод на *aplay* – **echo "Hello World" | festival_client --ttw | aplay**. Набрав ее в командной строке, вы должны услышать голос речевого синтезатора *Festival*. Теперь мы готовы всерьез взяться за программирование. »

› **Выбор пакетов *Festival* ошарашивает, но для получения рабочей установки все они не понадобятся.**

RSS против Atom

Мы уже привыкли к тому, что небольшой оранжевый значок, помечающий сгруппированные новости на сайте, обозначает ленту RSS. Однако есть много конкурирующих версий и форматов, которые прикрываются тем же символом. Два самых распространенных, RSS 2.0 и Atom, имеют полную поддержку со стороны модуля *Feedparser*, используемого в нашем сценарии. Atom был разработан с целью устранить некоторые проблемы RSS. В RSS, к примеру, не было способа указать, какой язык используется в конкретной истории, а также является ли она HTML или простым текстом. Но это несоответствие создает проблемы для разработчиков. Два разных формата необходимо и рассматривать по-разному, потому что Atom содержит элементы, которых нет в RSS, и даже форматирование общих элементов, типа даты и времени, заметно отличается. Нужно убедиться, что разбираемая лента поддерживает элементы в ожидаемом вами формате; единственный способ – загружать «сырые» данные ленты, либо скачав файл, на который указывает URL ленты, либо с помощью агрегатора новостей, извлекающего исходные данные и сохраняющего их на диске. Содержание ленты можно просмотреть в обычном текстовом редакторе.



› **Не все то канал RSS, что отмечено этой иконкой в вашем браузере.**

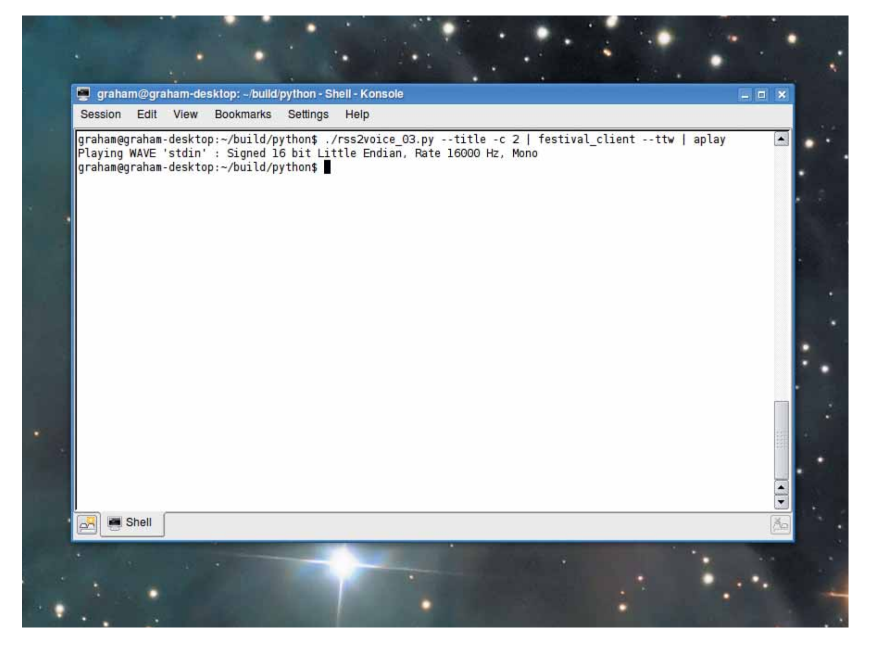
Скорая помощь

Не забывайте, что Python использует отступы для логического разделения участков кода, например, функций и условий **if**. Если вы получаете сообщения об ошибках, проверьте, все ли отступы используют одинаковое количество пробелов или знаков табуляции.

Часть 2 Декодируем RSS

Решив вопрос с генерацией звука, можно заняться программированием; начнем с загрузки и разборки RSS-ленты. Оно сперва звучит пугающе, но осуществляется очень легко, благодаря штуке под названием '*Feedparser* ' – модулю Python, который умеет загружать и разбирать все наиболее распространенные форматы лент RSS и Atom. Так как это модуль, его необходимо установить отдельно, но в силу его универсального великолепия он, скорее всего, поставляется вместе с Python в менеджере пакетов вашего дистрибутива. Если вы не пользовались Python раньше, он приятно поразит вас на фоне мира зависимостей, библиотек, заголовков и наследований, которыми страдают другие, более «корпоративные» языки программирования. Так, например, для импорта *Feedparser* в свою программу нужно просто начать ее с **import feedparser**. Только эта строка и нужна для поддержки RSS-лент в нашей Python-программе. Чтобы доказать это, мы добавим несколько строк, извлекающих заголовки первой ленты с нашего собственного сайта. Откройте ваш любимый текстовый редактор и добавьте следующий текст в файле ниже команды import:

› Беда скриптов, которые только издают звук, в том, что им нечего показать на экранном снимке. Придется вам самим представить, как пьяный голос синтезатора речи *Festival* сравнивает Windows с Ubuntu.




```
import feedparser
rss_url = "http://www.linuxformat.co.uk/backend.php"
feed = feedparser.parse( rss_url )
print feed.entries[0].title
```

Сохраните текст в файле с расширением **py** и запустите его из командной строки, набрав **python имяфайла.py**. Вы должны увидеть заголовок последней новости с сайта *Linux Format* UK, напечатанный в консоли. За это надо благодарить *Feedparser*: он скачивает ленту с **linuxformat.co.uk** и возвращает данные как Unicode-строку Python, которую мы связываем с переменной feed. Отсюда и происходят все другие данные RSS/Atom. В приведенном примере мы вывели заголовок первой новости. Каждая новость является элементом в массиве записей **entries**, и мы свободно можем запросить практически любое сообщение ленты. Заменив title на description, мы получили бы тело новости, а не один заголовок. Есть много элементов, подобных этим, и мы можем использовать их в зависимости от версии и формата канала.

Попробуйте добавить print feed для печати всего содержимого выбранного RSS-источника и поищите в выводе элементы, которые могут пригодиться. Общие элементы включают feed.title для заголовка ленты, feed.link, возвращающий URL сайта, и feed.date, содержащий дату ленты. Но мы еще раз предупреждаем, что все каналы отличаются друг от друга в своем представлении. Лента *Linux Format UK*, например, не включает поля даты. Чтобы сделать процесс более гладким, скрипты Python можно запускать автоматически; поместите это в первой строке исходного кода:

```
#!/usr/bin/python
```

Сделайте файл исполняемым с помощью

```
chmod +x filename.py
```

в терминале. Ваш скрипт теперь можно запустить, набрав **./filename.py**, и интегрировать его в вашу установку *Festival*, напечатав:

```
./filename.py | festival_client --ttw | aplay
```

Вы должны услышать заголовок нашей главной новости, читаемый вам синтезатором речи *Festival*, и это наиболее сложная часть всего процесса. Теперь можно адаптировать наш простой сценарий для выбора статей, которые вы хотите услышать, организовать поддержку нескольких лент и ввести любые другие улучшения. Но прежде всего нужно создать каркас для скрипта. Начнем снова с пустого файла и создадим более гибкое решение.

Часть 3 Программирование на Python

Вершиной хит-парада дружелюбия программы является умение разбирать параметры командной строки. Если мы добавим его в нашу программу, то сможем использовать различные аргументы для загрузки альтернативных лент, указания числа выводимых новостей для чтения, а также других режимов работы. С внедрением этих опций мы сможем автоматизировать нашу программу через *cron* или через другие сценарии, используя внутренние аргументы для изменения режима работы скрипта – точно так же, как мы делаем с другими утилитами. Анализ аргументов Python (так это называется) очень похож на используемый в языках C и C++ и использует модуль *getopt*. Его необходимо добавить к инструкции import в верхней части нашего скрипта, вместе с другим модулем по имени *sys* и ранее добавленным *Feedparse*. *Sys* обеспечивает базовую функциональность, зависящую от системы, и является обязательным для всех скриптов, кроме самых простейших. Теперь мы должны разделить наш сценарий на функции, чтобы мы могли вызывать каждую из них по отдельности в зависимости от аргументов командной строки. Как и в C/C++, мы начнем с функции main, где живет основная составляющая программной логики.

```
def main():
   try:
      opts, args = getopt.getopt(sys.argv[1:], "hu:c:t", ["help",
"url=", "count=", "title"])
   except getopt.GetoptError, err:
      print str(err)
      usage()
      sys.exit(2)
```

В первой строке функция main заявляет о своем существовании, def main():, затем разбирает аргументы командной строки внутри пары операторов try и except. Они представляют собой простейшую форму обработки исключений. Выражение с try анализирует переданные параметры, выискивая однобуквенные ключи h, u:, c: или t и их длинные аналоги help, url=, count= и title. Мы будем использовать эти аргументы, чтобы добавить функциональность в скрипт. Ключ -h или --help, например, заставит его выводить информацию об использовании, url="http://feedurl" сообщит скрипту адрес ленты для скачивания, а -c 3 или --count=3 велят ему принять во внимание только три послед-

них новости. Наконец, аргументы -t and --title выступают в качестве переключателя. Если они присутствуют, скрипт будет выводить только заголовок каждой новости, а не заголовок вместе с текстом. Если какие-либо аргументы не распознаны, запускается сегмент исключения, выводя стандартную ошибку перед выполнением функции usage() и выходом. Теперь, определив и разобрав аргументы командной строки, используемые для запуска скрипта, мы должны сделать с ним что-нибудь полезное. Используя цикл for, переберем аргументы командной строки и назначим их значения переменным, которые мы сможем использовать в части скрипта, отвечающей за работу с RSS. Ниже обработчика исключений добавьте следующий кусок кода:

```
url = "http://www.linuxformat.co.uk/backend.php"
 title = False
 count = 1
 for o, a in opts:
    if o in ("-h", "--help"):
       noargument()
       sys.exit()
    elif o in ("-u", "--url"):
       url = a
    elif o in ("-c", "--count"):
       count = int(a)
    elif o in ("-t", "--title"):
       title = True
    else:
       assert False, "invalid option"
```

Мы создаем значения по умолчанию для трех переменных, которые будем использовать: url для месторасположения ленты, title для выбора, следует ли включать описание каждой новости или просто вывести название, и count для указания количества читаемых новостей.

В условиях if мы проверяем каждую опцию и аргумент командной строки (o и a в коде). Если определены -h или --help, будет вызвана функция noargument, и скрипт завершит работу. Если определены -u или --url=, адрес ленты (аргумент) будет присвоен переменной url. При обнаружении -c или --count=, функция count = int(a) преобразует строковый аргумент в целое число и запишет его в переменную count. Наконец, если будут найдены -t или --title, то флаг title будет установлен в **True** (истина).

Понять смысл этих настроек очень легко с помощью кода *Feedparser*, использованного в нашем первоначальном скрипте. Мы просто используем еще один цикл for для извлечения каждой новости в зависимости от значения переменной count (которая по умолчанию равна **1**), и решаем, брать ли только описание новости или заодно и ее содержание. Вот код для этого – он должен идти следом за предыдущим куском:

```
feed = feedparser.parse( url )
feed['feed']['title']
for i in range(0, count):
   if title:
      print feed['entries'][i]['title']
   else:
      print feed['entries'][i]['title']
      print feed['entries'][i]['description']
```

Здесь мы просто обрабатываем каждую новость в соответствии со значением count и передаем текст на стандартный вывод. Это будет терминал, и уже оттуда можно перенаправить текст в синтезатор речи *Festival*. Вот и все, что надо включить в функцию main. Осталось только чуть-чуть причесать код. Под нашим фрагментом кода добавьте следующее дополнение, необходимое для любого скрипта Python с функцией main():

```
if __name__ == "__main__":
   sys.exit(main())
```

Взглянув на вызываемые нами функции, вы заметите, что используются две еще неописанные: usage(), ответ на запуск обработчика исключений при отсутствии правильных аргументов, и noargument, вызываемая, когда пользователь запрашивает --help. Обе они должны слегка подтолкнуть пользователя в нужном направлении, предлагая ему, во-первых, ввести скрипт с опцией --help, а во-вторых, описывая каждый из ключей, принимаемых скриптом. Для выполнения обеих функций добавьте нечто похожее на следующий код в верхнюю часть вашего скрипта, между оператором import и началом функции def main():.

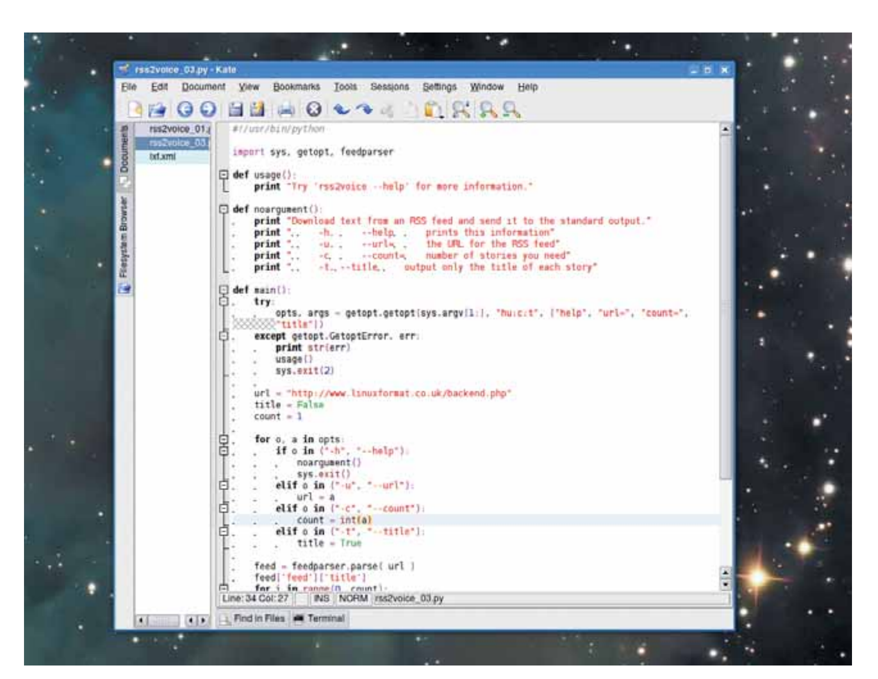
```
def usage():
   print "Try 'rss2voice --help' for more information."

def noargument():
   print "Download text from an RSS feed and send it to the
standard output."
   print "-h, --help prints this information"
   print "-u, --url=the URL for the RSS feed"
   print "-c --count= number of stories you need"
   print "-t,          --title output only the title of each story"
```

Вот и весь код. Вы сейчас написали полностью функциональный сценарий разбора ленты RSS/Atom, пригодный для отправки текста в синтезатор речи *Festival*. Как обычно, при ограниченном пространстве для публикации, мы опустили проверку на ошибки – хуже всего то, что мы не проверяем правильность URL'а, передаваемого в скрипт. Если ленты не существует, это приведет к проблемам, и может вызвать ошибки в скрипте. В остальном все готово к работе. Просто замените новым скриптом старый в первоначальной команде, которую мы использовали для озвучивания речи в *Festival*, и вы услышите мелодичные голоса, рассказывающие вам последние новости из Башен *Linux Format*. LXF

Куда двигаться дальше...

- Создать запись *cron* для автоматического запуска скрипта.
- Добавить режим ожидания для просмотра новых историй каждый час.
- Отмечать прочитанные и непрочитанные статьи, чтобы избежать дублирования.
- Настроить различные синтезаторы речи или голоса.
- Интегрировать речь *Festival* непосредственно в скрипт.



› Текстовый редактор KDE, *Kate*, подсвечивает синтаксис Python и отмечает границы циклов и функций.

» Через месяц Создадим графический интерфейс для синтезатора речи *Festival*.



Новая серия! Создаем приложения для самого современного рабочего стола в Linux.





KDE 4

ЧАСТЬ 1 Считаете, что для KDE 4 мало приложений? В ваших силах исправить ситуацию – благодаря новой серии учебников от **Андрея Боровского!**

Два года назад **LXF** публиковал цикл статей о программировании для KDE 3 (LXF81—85). Тогда это был зрелый рабочий стол, который готовился уступить место KDE 4, и писать о нем было легко, не не слишком интересно. Легко – поскольку KDE 3 на тот момент был хорошо (даже очень хорошо) документирован, а также потому, что существовало много примеров (в том числе написанных мною самим). С другой стороны, было понятно, что цикл разработки KDE 3 практически завершен, и в статьях о нем не удастся отразить живые моменты рождения новых технологий.

С KDE 4 все обстоит иначе. Его первый стабильный релиз появился в начале этого года, и сейчас KDE 4 активно развивается. Далеко не все идеи разработчиков получили окончательное воплощение; по-видимому, и сам список технологий, используемых в KDE 4, будет со временем меняться – я, например, не уверен, что система *Decibel*, о которой будет сказано ниже, войдет в последующие релизы. Что же касается документации и примеров, то стоит ли говорить, что для многих частей KDE 4 и то, и другое существует либо в зачаточной стадии, либо вовсе отсутствует. Именно поэтому писать о KDE 4 гораздо интереснее. У нас есть возможность проследить глазами программиста рождение нового рабочего стола, с которым пользователи Linux будут жить несколько ближайших лет. Как и в случае с *Qt 4*, мы будем предполагать, что вы уже знакомы с разработкой для KDE 3. На всякий случай, уроки в формате PDF есть на DVD этого номера.

А внутри у ней плазма

Библиотеки *Qt 4.x* во многом революционны, и было бы странно, если бы разработчики KDE 4 не подхватили этот «трудовой порыв». На внешних красотах KDE 4, усиленных мощью 3D-ускорителей, мы останавливаться не будем, а скорее поднимем капот (или снимем кожух?) и посмотрим, что же находится внутри. Один из важнейших элементов KDE, который сразу бросается в глаза (в нескольких смыслах слова) – это Plasma. С точки зрения программиста, она – развитие Qt Graphics View Framework (LXF105). Система *Corona* расширяет возможности QGraphicsScene, добавляя концепцию апплетов. В результате с элементами графической сцены (т.е. рабочего стола) можно связывать мини-приложения. Изначально разработчики Plasma ориентировались на *Qt 4.2*, в которой, напомню, еще не было встраиваемых виджетов; в результате пришлось придумывать аналогичный механизм для внедрения пиктограмм и кнопок. Теперь, когда данный функционал появился в самой *Qt*, можно ожидать более широкого использования встраиваемых виджетов в будущих версиях Plasma.

Минуя другие состояния вещества, перейдем от Plasma к технологии Solid. Это – кросс-платформенная подсистема, предназначенная для работы с устройствами. Правда, Solid не позволяет KDE-приложениям управлять ими, но предоставляет достаточно информации, чтобы программа могла подобрать нужные административные средства самостоятельно.

Еще одна интересная и пока что довольно загадочная новинка KDE 4 – Phonon. Это переносимая мультимедиа-подсистема, поставляемая с *Qt* начиная с версии 4.4. API Phonon состоит из классов высокого уровня, которые унифицируют доступ к различным библиотекам, связанным с мультимедиа. Судя по документации (которая на момент написания статьи была еще не завершена), важную роль в Phonon играют концепция прозрачного доступа к мультимедиа-ресурсам, независимо от того, где они расположены, и унаследованная от *Qt* система «модель-контроллер-вид».

Технология под названием *Decibel* должна упростить жизнь разработчикам IP-телефонов, видеофонов, чатов и т.п. Она написана с использованием чистого *Qt 4* (т.е. не зависит от KDE), включает в себя реализацию *Telepathy* и многих других полезных вещей, но, судя по сайту проекта (decibel.kde.org), работа над ним застопорилась примерно год назад. Текущая версия – 0.5, что будет дальше – посмотрим.

Инструменты разработчика

Чтобы программировать для KDE 4, нам, конечно, понадобится сама среда. Проще всего воспользоваться дистрибутивом, в котором KDE 4 доступен по умолчанию, но и тут могут возникнуть неожиданные проблемы. Например, в «живом» OpenSUSE 11.0 KDE Four Live с KDE 4.1 Beta2 обнаружилась одна серьезная и надоедливая ошибка – любое приложение KDE 4, выводящее стандартное диалоговое окно, вызывает фатальный сбой в *KNotify* (о чем тут же сообщает еще один диалог). Поэтому нам придется довольствоваться KDE 4.04, входящим в состав OpenSUSE 11.0. Помимо собственно графической среды, нам, разумеется, понадобятся пакеты разработчика *Qt 4.x* и KDE 4.x.

На момент написания этой статьи стабильного *KDevelop 4* еще не существовало, а нестабильный работал только с тестовой версией KDE 4.1, от использования которой мы с вами отказались. В Сети можно найти трюки, с помощью которых можно создавать приложения KDE 4 в *KDevelop 3.x*, но мы на них останавливаться не будем: думаю, что *KDevelop 4* уже скоро обретет стабильность, и данные советы морально устареют. Лучше потратим сэкономленную журнальную площадь на *KDevelop 4*, когда он выйдет. Пока же будем применять инструменты низкого уровня: это позволит нам глубже проникнуть в суть вещей в KDE 4. Не считая стандартных инструментов разработчика, нам понадобятся две консольные утилиты: *cmake* и *kapptemplate* – мастер, с вызова которого и начинается процесс разработки. *Kapptemplate* создает заготовки для простого приложения KDE, приложения KDE со сложным главным окном или *KPart*.

под микроскопом

Далее утилита просит указать имя и расположение нового проекта. Программа также захочет «познакомиться» – узнать ваши имя и e-mail. В этом смысле *kapptemplate* отличается крайней навязчивостью и задает личные вопросы при каждом запуске.

Директория, в которой *kapptemplate* сохраняет файлы заготовки программы, содержит много интересного (об этом – чуть позже), но в ней отсутствуют Make-файл и сценарий *configure*, то есть те скрипты, которые мы привыкли использовать для сборки программ KDE 3. Здесь нам на помощь приходит *CMake*. Этот кросс-платформенный независимый инструмент от компании Kitware стал частью инструментария KDE 4 совсем недавно. Утилита *cmake* выполняет для проектов KDE (и не только KDE) ту же работу, что *qmake* для *Qt* – т.е. создает Make-файл проекта. *Cmake* вызывает *qmake* по ходу процесса, и вы должны убедиться, что утилита с таким именем в вашей системе относится к *Qt 4*, а не *3.x*. Для этого наберите в окне консоли:

```
qmake --version
```

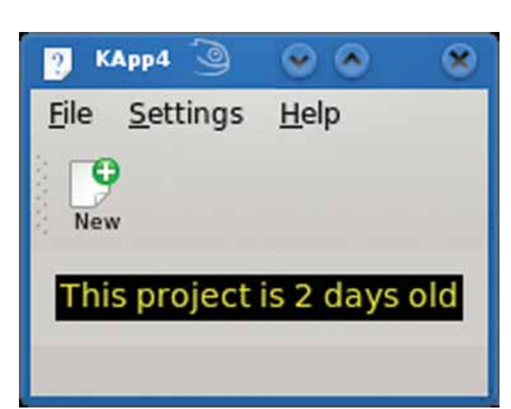
Если *qmake* «не тот», ситуацию следует исправить с помощью переменной PATH. В самом простом случае (который мы и рассматриваем) для генерации Make-файлов для проекта KDE 4 достаточно скомандовать

```
cmake <project_path>
```

где project_path – это путь к проекту. Имейте в виду, что *cmake* создает выходные файлы, в том числе и Makefile, не в project_path, а в той директории, откуда он был вызван. Теперь, наконец, можно скомандовать

```
make
```

и насладиться окном нашего первого приложения KDE 4 (**рис. 1**). Что же это за зверь такой, сурово указывающий на несвежесть файлов проекта?



› Рис. 1. Минималистичное приложение KDE 4.

Через увеличительное стекло

Если ваша программа называется *kapp4*, то объявление класса главного окна содержится в файле **kapp4.h** (а его методы – в **kapp4.cpp**). Класс, реализующий центральный элемент главного окна – KApp4View – располагается в файлах **kapp4view.h/kapp4view.cpp** и наследует от Ui::kapp4view_base, объявление которого (файл **ui_ kapp4view_base.h**) генерируется автоматически на основе описания интерфейса программы, содержащегося в **kapp4view_base.ui**. В этой структуре нельзя не заметить влияния *Qt 4.x*, и (как вы, наверное, уже догадались) класс kapp4view_base.ui не является потомком QWidget, но содержит инструкции, позволяющие настроить таковой (LXF104). Помимо **kapp4view_base.ui**, в нашем проекте есть еще один ui-файл – **prefs_base.ui**, который содержит описание интерфейса окна настроек. Забегая вперед, отмечу, что в KDE 4 вы «бесплатно» получаете мощную систему конфигурации приложения пользователем (правда, все равно уступающую таковой в программе-часах из LXF106).

Вы, наверное, уже озадачились вопросом: как редактировать ui-файлы, если *KDevelop* отсутствует? Отвечаем: для визуального проектирования интерфейса можно воспользоваться редактором *Qt Designer 4.x* – он поддерживает и виджеты KDE 4. Файлы **settings.h** и **settings.cpp** содержат, соответственно, объявление и определение класса Settings. Он наследует от KConfigSkeleton и является основой движка настройки нашего приложения. Наконец, файл **main.cpp**,

Проект CMake: что внутри?

Если даже раньше вы никогда не работали с *CMake*, то после использования данной утилиты должны почувствовать, насколько она удобна. Создать Make-файлы для стандартного проекта с помощью *CMake* просто, но вы наверняка хотите знать, как можно влиять на этот процесс – например, добавить ссылки на разделяемые библиотеки? В *qmake* для этого потребовалось бы отредактировать файл ***.pro**, а его аналогом в *CMake* служит **CMakeLists.txt**. Синтаксис **CMakeLists.txt** довольно сложен, поэтому мы рассмотрим только самые важные элементы. Каждая строка файла содержит команду и ее аргументы, перечисляемые в круглых скобках. Так, команда project() указывает имя проекта. Здесь можно также задать язык программирования: C, CXX, Java. Команда find_package() находит и загружает данные внешнего пакета. Например,

```
find_package(KDE4 REQUIRED)
```

указывает, что требуется загрузить пакет KDE4. Опция REQUIRED уточняет, что без него сборка проекта невозможна (логично, не правда ли?). После загрузки пакета KDE4 файлу настройки *CMake* становятся доступны определенные в нем переменные и команды. Например, команда Include_directories() использует переменные KDE4_INCLUDES и QT_INCLUDES. Команда kde4_add_ui_files() не относится к числу встроенных в *CMake* – она также добавляется пакетом KDE4.

Команда target_link_libraries() (на этот раз – стандартная) сообщает, какие библиотеки следует подключить к данной цели сборки (она задается первым аргументом). Команда получает имена библиотек из переменной KDE4_KDEUI_LIBS, которая определена в пакете KDE4.

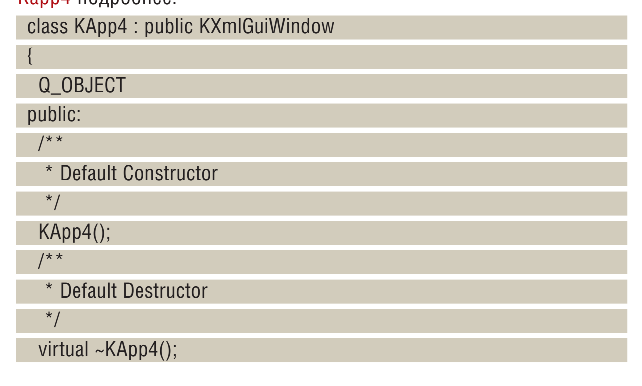
С помощью файла **CMakeLists.txt** настоящие мастера могут творить чудеса, однако если премудрости его синтаксиса вас пугают, можете утешиться тем, что вам вовсе не обязательно знать их все. Если вы используете мастер генерации проектов KDE 4, то по большей части настройка файла **CMakeLists.txt** будет сводиться к незначительному редактированию уже имеющейся

как нетрудно догадаться, предоставляет функцию main() нашей программы.

При анализе исходных текстов мы будем двигаться сверху вниз, т.е. от main() к различным классам. Если отбросить ненужные нам в настоящий момент элементы, то функцию main() можно свести к следующей схеме:

```
int main(int argc, char **argv) {
  KApplication app;
  KApp4 *widget = new KApp4;
  widget->show();
  return app.exec();
}
```

Класс KApp4 реализует главное окно нашего приложения. Мы не сообщаем объекту app класса KApplication, что widget указывает на главный визуальный объект программы. Рассмотрим объявление Kapp4 подробнее:



```
class KApp4 : public KXmlGuiWindow
{
  Q_OBJECT
public:
  /**
   * Default Constructor
   */
  KApp4();
  /**
   * Default Destructor
   */
  virtual ~KApp4();

private slots:
  void fileNew();
  void optionsPreferences();

private:
  void setupActions();
private:
  Ui::prefs_base ui_prefs_base ;
  KApp4View *m_view;
  QPrinter   *m_printer;
  KToggleAction *m_toolbarAction;
  KToggleAction *m_statusbarAction;
};
```

Класс главного окна является потомком KXmlGuiWindow, который происходит от класса KMainWindow, наследующего от QMainWindow, наряду с KXmlGuiBuilder и KXmlGuiClient. Ниже приводится листинг конструктора класса главного окна

```
KApp4::KApp4()
  : KXmlGuiWindow(), m_view(new KApp4View(this)), m_printer(0) {
   // accept dnd
   setAcceptDrops(true);
   // tell the KXmlGuiWindow that this is indeed the main widget
   setCentralWidget(m_view);
   // then, setup our actions
   setupActions();
   // add a status bar
   statusBar()->show();
   // a call to KXmlGuiWindow::setupGUI() populates the GUI
   // with actions, using KXMLGUI.
   // It also applies the saved mainwindow settings, if any, and ask the
   // mainwindow to automatically save settings if changed: window size,
   // toolbar position, icon size, etc.
   setupGUI();
}
```

Я специально не стал убирать из листинга комментарии, добавленные в него программой-генератором, чтобы показать, что файлы-заготовки, созданные *kapptemplate*, не только освобождают нас от рутинной работы, но и сообщают полезную для новичков информацию о том, что делают вызываемые функции. Для настройки внешнего вида главного окна, как и в KDE 3, используется система KXMLGUI. Напомню, что главное окно хранит данные о меню, панелях инструментов и командах в файлах ***ui.rc**, имеющих формат XML. Для нашего приложения тоже был создан такой – это **kapp4ui.rc**, расположенный в той же директории, что и исполняемый файл. Тем не менее, при каждом запуске программы на консоль будет выводиться сообщение о том, что файл **kapp4ui.rc** не найден (это не приводит к завершению программы, поскольку, не найдя свой собственный rc-файл, приложение берет стандартный из KDE 4). Наша программа не может найти файл **kapp4ui.rc** потому, что ищет его не там, где он лежит. Глобально установленные приложения, доступные всем пользователям, хранят свои файлы ***ui.rc** в общедоступной директории (например, **/usr/share/kde4/apps/имя_приложения**). Для сохранения ваших персональных настроек следует воспользоваться каталогом **~/.kde4/share/apps/имя_приложения**.

Система настройки

Выше я уже упоминал о том, что стандартный проект приложения KDE 4 по умолчанию снабжается системой настройки. Ее можно разделить на две части: настройка панелей инструментов, сочетаний горячих клавиш – с одной стороны, и конфигурация специфических параметров работы приложения – с другой. Особенность настройки первого типа заключается в том, что, по существу, она одинакова для всех приложений KDE и может быть легко формализована. Однако разработчики KDE на этом не успокоились и сделали все возможное, чтобы охватить и процесс создания средств специфической настройки приложений. Окно специальных настроек (**рис. 2**) открывается по команде Configure KApp4... меню Settings.

Как мы видим, здесь можно настроить цвет шрифта и фона рабочей области главного окна (стыдно называть этот объект рабочей областью, ведь он ничего не делает!) и возраст проекта в днях. В результате внешний вид программы может довольно сильно измениться (**рис. 3**).

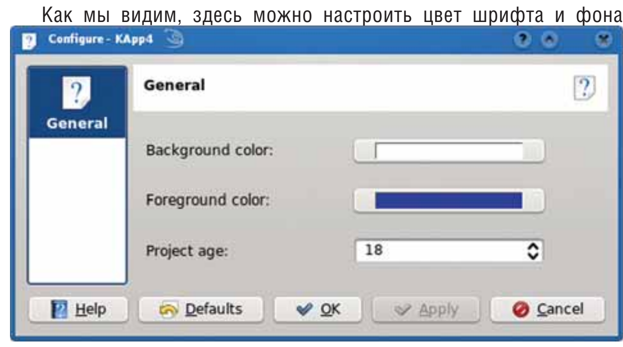


Рис. 2. Окно настройки программы *KApp4*.

Глобальные установки хранятся в директории **/usr/share/kde4/**

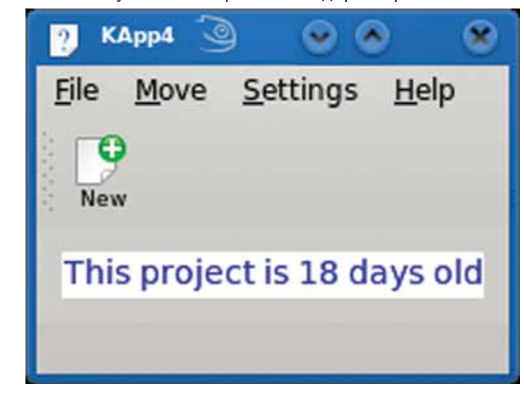


Рис. 3. Смена оформления *KApp4*.

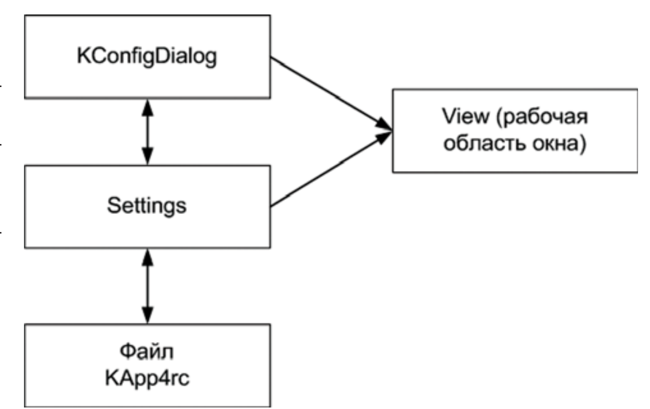
config, а локальные – в **~/.kde4/share/config**. После нескольких запусков программы *KApp4* я нашел в ней два файла: **KApp4rc**, где хранятся параметры рабочей области, и **kapp4rc**, содержащий данные о геометрии главного окна и видимости его элементов. Следует отметить, что многие программы KDE 4 хранят свои параметры конфигурации сразу в нескольких файлах.

Для управления настройками создается глобальный объект класса Settings. Получить указатель на него можно с помощью статического метода Settings::self(). Рассмотрим фрагмент исходного текста класса KApp4, управляющий созданием диалогового окна настроек:

```
KConfigDialog *dialog = new KConfigDialog(this, "settings",
Settings::self());
QWidget *generalSettingsDlg = new QWidget;
ui_prefs_base.setupUi(generalSettingsDlg);
dialog->addPage(generalSettingsDlg, i18n("General"), "package_setting");
connect(dialog, SIGNAL(settingsChanged(QString)), m_view,
SLOT(settingsChanged()));
dialog->setAttribute( Qt::WA_DeleteOnClose );
dialog->show();
```

Здесь создается объект dialog класса KConfigDialog, который реализует диалоговое окно. Конструктору класса передается указатель на объект Settings. Далее создается объект generalSettingsDlg класса QWidget. Он должен стать главной страницей окна настроек. Внешний вид объекта generalSettingsDlg задается методом ui_prefs_base.setupUi() (так же, как и в *Qt 4*). Затем, с помощью метода addPage() новая страница добавляется в окно настроек. Остается только связать сигнал settingsChanged объекта dialog с одноименным слотом объекта m_view (рабочая область окна программы). Обработчик сигнала settingsChanged может извлекать данные об изменениях настроек из строки, переданной ему в качестве параметра, или непосредственно из объекта Settings, который существует глобально (именно так и поступает слот settingsChanged() в рассматриваемой программе). Взаимоотношения между окном KConfigDialog, объектом Settings и рабочей областью окна можно проиллюстрировать графически (**рис. 4**).

Скорее всего, разработчики KDE 4 почувствовали то же, что чувствуете сейчас вы: программирование всех этих классов настройки вручную слишком утомительно. Поэтому они решили автоматизировать процесс. Для создания класса Settings и его вспомогательного класса была использована специальная консольная утилита – *kconfig_compiler*. На самом деле это, конечно, не компилятор, а генератор исходных текстов, которые потом компилируются GCC. Исходниками для *kconfig_compiler* являются два файла – **kapp4.kcfg** и **kapp4.kcfgc**.



› Рис. 4. Система настройки программы KApp4.

Если вы откроете файл **kapp4.kcfg** в стандартном окне просмотра *Dolphin* (мы ведь работаем в KDE 4, не так ли?), то увидите только невразумительное:

```
color of the background black color of the foreground yellow size of a
ball 2
```

Дело в том, что **kapp4.kcfg** – файл в формате XML. Чтобы понять, как он выглядит на самом деле, нужно выбрать команду View Document Source. Основные тэги файла ***.kcfg** – это <kcfgfile>, <group> и <entry>. Тэг <kcfgfile> принимает в качестве параметра имя файла, где будут сохраняться настройки программы. Тэг <group> описывает группу настроек. Его дочерние элементы – тэги <entry> – представляют собой отдельные настройки (имя переменной, в которой будет храниться значение конфигурационного параметра, тип данных, имя настройки, предназначенное для пользователя и значение по умолчанию). Кроме того, файлы ***.kcfgc** содержат указания для генератора *kconfig_compiler*. Их можно сравнить с Make-файлами GCC.

Что ж, для первого раза, наверное, достаточно. Загляните в те файлы, на которые нам не хватило места, а если вы достаточно храбры – попробуйте модифицировать программу, вооружившись только что полученными знаниями. LXF

» Через месяц Мы подробно рассмотрим самую наглядную из технологий KDE 4 – Plasma.





Имитационное моделирование

ЧАСТЬ 3 Одним из великих даров компьютерной эры является возможность увидеть невидимое и услышать неслышимое. **Александр Бикмеев** предлагает заглянуть в «черный ящик».

Не знаю, как вам, а мне всегда хотелось узнать и увидеть недоступное: как выглядит черная дыра изнутри, где находится граница литосферы и магмы, на что была похожа Вселенная за мгновение до Большого взрыва... да хотя бы что происходит с шариком (или мухой) в закрытой коробке, когда я ее трясу? Благодаря *Scilab* и моим Учителям я могу проникнуть в эти тайны, но рамки журнала не позволяют рассказать обо всем. Поэтому придется проявить скромность и остановиться на коробке и шарике.

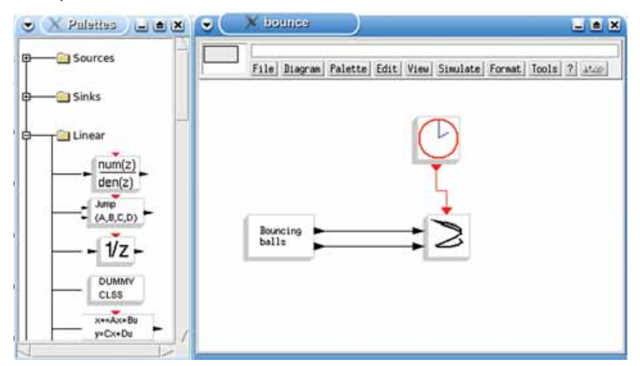
То, чем мы собираемся заняться, называется компьютерным экспериментом в форме имитационного моделирования. Для этой цели в системе *Matlab* есть *Simulink*, а в *Scilab* – *Scicos*. Это инструмент для моделирования динамических процессов с помощью функциональных блоков и связей между ними. Проще говоря, математическая модель изучаемого объекта или процесса составляется из определенных блоков, а затем при помощи связей указывается, как данные перемещаются от одного блока к другому. Да, то же самое можно проделать, написав программу на встроенном языке, но если уже есть готовые блоки, так почему бы не воспользоваться ими? Кроме того, описание модели в виде диаграммы более наглядно и информативно.

Конечно, *Scicos* содержит не так много диаграмм, как *Simulink*, однако для моделирования многих объектов и процессов, а особенно для целей обучения он вполне подойдет.

Ну-с, приступим

Scicos содержит графический редактор, представляющий собой окно, в котором отображается диаграмма и отдельные окна палитр блоков. Вызывается он командой **scicos;** (точку с запятой можно не ставить, но тогда все окно *Scilab* будет заполнено служебной информацией). По сути, окно графического редактора *Scicos* не очень-то и отличается от самого *Scilab*, но в нем нельзя вводить текст; вы также заметите, что главное меню содержит несколько новых пунктов: Diagram (для задания параметров всей диаграммы), Simulate (для компиляции, определения параметров и запуска модели), Tools (инструменты для различных служебных задач) и Palette (работа с палитрами функциональных блоков). Это графическое окно, и если белый фон навевает на вас скуку, можете изменить его, щелкнув правой кнопкой мыши в пустой области окна диаграммы и выбрав в появившемся меню пункт Background color.

Все блоки сгруппированы в палитры. Вывести любую из них можно, выбрав последовательно Palette > Palettes, а затем щелкнув на требуемой панели в появившемся окне. Однако более удобно в работе древо палитр, поскольку в этом случае все блоки присутствуют на экране одновременно (см. **рис. 1**): оно вызывается последовательным выбором Palette > Pal tree.



› Рис. 1. Так выглядит графический редактор *Scicos* и древо палитр блоков.

Блоки могут иметь различное число входов и выходов, называемых *портами*. Входной порт обозначается треугольником с углом, направленным в блок, а порт-выход – с углом, направленным из блока. Порты делятся на обычные (для ввода/вывода данных в/из блока), которые отображаются черным цветом, и *активационные* или *управляющие* (для ввода/вывода управляющей информации), отмечаемые красным цветом. Как правило, обычные порты расположены по бокам блока, а управляющие – сверху и снизу.

При составлении диаграммы вначале необходимо дважды щелкнуть на желаемом блоке, а затем перенести курсор мыши (он примет вид иконки блока) в окно диаграммы и левым щелчком установить блок в нужном месте. Позиция блока не фиксирована, то есть впоследствии, если потребуется, вы сможете переместить блок, не нарушая связи и сохранив его атрибуты. Блоки также можно поворачивать и отражать: чтобы сделать это, щелкните на блоке правой кнопкой мыши и выберите в появившемся меню один из трех пунктов: Rotate Left (повернуть против часовой стрелки на 45^0), Rotate Right (повернуть по часовой стрелке на 45^0) или Flip (перевернуть, отразить).

После переноса блоков их необходимо соединить между собой. Для создания связи следует щелкнуть на треугольнике выходного порта одного блока, а затем – на треугольнике входного порта друго-

» Месяц назад Мы разбирались с внутренними и внешними функциями *Scilab*.

го. При этом будет создана прямая линия. Если вам необходимо создать ломаную, то каждую ее точку можно определить левым щелчком в пустой области, точно так же, как при рисовании многоугольника в *OOo Draw*. Когда вы, наконец, щелкнете на входном порту, редактор постарается сделать углы вашей линии прямыми. Каждый входной порт соединяется лишь с одним выходным портом. Если вам необходимо провести линии от одного блока ко многим, например, передать одно и то же значение на вход нескольких расчетных блоков или синхронизировать несколько генераторов, то сначала проводится одна связь, а затем от ее линии создаются ответвления. Для этого следует выполнить двойной щелчок в каком-либо месте связи и провести дополнительную ветвь. Альтернативный вариант – использовать блоки MUX и DEMUX.

Существует два вида связей: для передачи данных (черного цвета) и для передачи управляющей/временной информации (красного цвета). Пользователь может изменить цвета связей, но не цвет входа/выхода блока: он четко указывает необходимый тип связи. Следует также помнить, что обычный и управляющий порты соединить нельзя.

Почти все блоки имеют изменяемые параметры, определяющие обработку информации или поведение. Чтобы открыть окно с перечнем параметров, достаточно дважды щелкнуть на блоке левой кнопкой мыши.

Как создаются диаграммы, надеюсь, понятно. Но как же выбирать блоки для решения конкретной задачи и какие блоки с чем соединять? Вот в этом-то и заключается искусство экспериментатора. В рамках данной статьи я не смогу описать назначение всех блоков (читайте справку), но постараюсь объяснить базовые принципы.

Что в черном ящике?

Прежде чем переходить к созданию диаграммы, следует построить математическую модель, то есть записать уравнение или систему уравнений, описывающих процесс, а также определить начальные условия и внутренние константы. *Scicos* сможет решить поставленную перед ним задачу только при условии, что вы сами понимаете ее, в противном случае не поможет и суперкомпьютер.

Рассмотрим такой пример: в коробке находится несколько упругих шариков, и в какой-то момент времени мы начинаем наблюдать за ними. Для этого случая модель уже построена и включена в демонстрационные примеры, поэтому в редакторе *Scicos* выберите последовательно пункты меню ? > Demos, а затем, в появившемся окне, укажите файл **bounce.cosf**. После этого в редакторе откроется диаграмма, показанная на **рис. 1**. В ней всего три блока, так что разобрать ее труда не составит.

Первый (имеющий вид красных часов) – это стандартный управляющий блок, без которого не обходится практически ни одна диаграмма. Он генерирует регулярные события, с заданным пользователем периодом. В данном случае он отсчитывает время, то есть подает команду на пересчет системы и отрисовку текущего состояния через определенные интервалы. Этот блок имеет всего два параметра: Period – период генерации событий и init time – начальное значение времени.

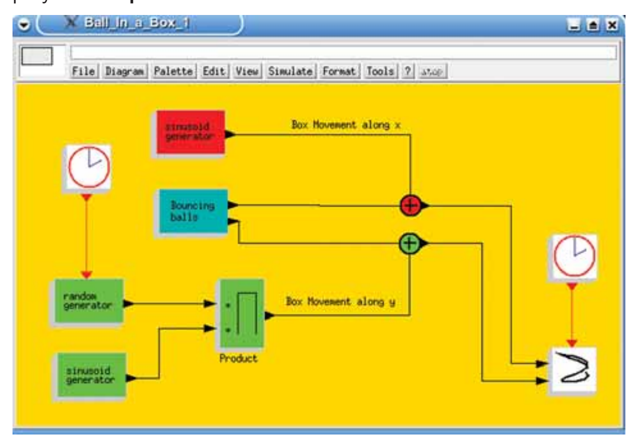
Блок, с которым соединены часы, представляет собой инструмент анимации, подстроенный специально для данной задачи. (Есть сходный инструмент, расположенный на палитре Sinks.) У него много параметров, и большая их часть задается векторами, размерность которых совпадает с числом шариков. Они определяют цвета шариков, их радиусы и границы коробки.

Ну, а третий блок осуществляет расчеты и выдает вектора координат. У него тоже много параметров, которые задают массу, радиус шариков, коэффициент аэродинамики, ускорение свободного падения...

Запустите модель на выполнение (Simulate > Run), и вы увидите, что происходит в неподвижной коробке с несколькими шариками. Хотите почувствовать разницу между Землей, Луной и Юпитером? Измените параметр g (gravity) и посмотрите, как он влияет на поведение системы. Чтобы остановить выполнение, нажмите на кнопку Stop в редакторе диаграмм (заметьте – не в окне с анимацией).

Но в нашей задаче всего один шарик, и коробка движется. Попробуем изменить данную модель в соответствии с новыми условиями. Пусть коробка (и шарик) находятся в поле тяготения и коробка совершает синусоидальные колебания вдоль горизонтальной оси, а также периодические со случайной амплитудой – вдоль вертикальной.

Добавим к открытой диаграмме описание смещения коробки, то есть синусоидальное изменение координаты шарика *x* и синусоидальные изменения со случайной амплитудой координаты *y*. Посмотрите результат на **рис. 2**.



› Рис. 2. Измененная диаграмма с заданным смещением коробки.

Блок с красным фоном описывает смещение коробки по оси *x* и представляет собой генератор синусоидального сигнала с палитры Sources. У него есть три параметра: Magnitude – максимальная по модулю величина сигнала, в нашем случае это максимальное смещение коробки (установите **2**); Frequency – частота колебаний (мы выберем здесь значение **15**); и phase – начальная фаза колебаний (укажите **0**).

Блоки с зеленым фоном задают смещение коробки по вертикали. Первый из них – это просто генератор событий (красные часы), который используется для выдачи случайного значения амплитуды колебаний (установите значение period в **0.2**). Далее идет сам блок-генератор случайных чисел (random generator). У него есть несколько интересных нам параметров, например, flag – флаг, определяющий как использовать параметры A и B:

» 0: A – это минимум, а максимальное значение равно A + B;

» 1: A – это среднее, а B – отклонение от среднего.

Установите flag равным **1**, A – **0**, B – **0,5**. Ниже расположен знакомый нам генератор синусоиды. Укажите в нем следующие значения параметров: Magnitude – **0.5**, Frequency – **5**. Затем случайное значение амплитуды умножается на периодический сигнал (блок Product панели NonLinear) и прибавляется к координате *y*.

Теперь нам осталось задать параметры расчета смещения шарика. Дважды щелкаем на блоке Bouncing balls и устанавливаем следующие значения: Mass – **[1]**, radius – **[1]**, [xmin,xmax,ymin,ymax] – **[0;10;0;10]**; xpos, xdpos, ypos, ydpos – **5**. Файл диаграммы со всеми изменениями (**Ball_in_a_box1.cos**) вы можете найти на **LXFDVD** в разделе **Журнал**.

Чтобы посмотреть движение нескольких шариков, необходимо изменить параметры блока Bouncing balls, указав для Mass, radius, xpos, xdpos, ypos, ydpos вектора с числом элементов, равным количеству шариков. Но этого мало: необходимо также добавить в диаграмму перед блоками суммирования блок Scalar to vector с палитры Branching. Ведь смещение коробки вдоль каждой оси есть скаляр, а координаты нескольких шариков передаются в виде векторов. Измененную диаграмму для этого случая (файл **Ball_In_a_Box_1m.cos**) вы также сможете найти на диске.

Запустите модель на выполнение. Правда, похоже? Но шарик вылетает за стенки коробки, а это неправильно! Так происходит

Немного истории

Scicos – это часть пакета *Scilab*, распространяемая совместно с ним. Изначально она разрабатывалась Р. Никухой [R. Nikoukhah] на языке *Scilab* и не имела графического интерфейса. Позднее, для первого официального релиза в 1994, разработчик переписал имитатор на Fortran и частично на C, а С. Штеер [S. Steer] разработал графический редактор. До конца 90-х *Scicos* развивался медленно силами студентов и аспирантов. С 2000 года работа над ним значительно ускорилась благодаря контракту с R&D. В частности был разработан новый имитатор на C, расширены типы данных, внедрен компилятор Modelica, разработаны интерфейсы взаимодействия с внешними устройствами.

Ключевую роль в продвижении *Scicos* сыграла книга “Modeling and Simulation in Scilab/Scicos”, вышедшая в издательстве Springer в 2006 году (к сожалению, на русский язык она не переведена). В 2008 году для версии *SciLab 4.1.2* был разработан абсолютно новый многооконный графический интерфейс и алгоритм компиляции, кроме того, *Scicos* был адаптирован под лицензию GPL.

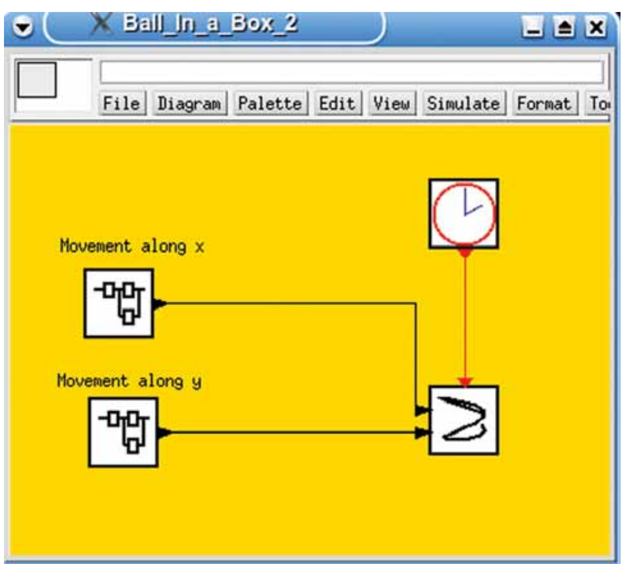
потому, что мы добавили смещение коробки к координатам уже при выполнении анимации, а на расчет наши добавки никак не влияют.

Таким образом, готовая модель не помогла нам точно описать нашу систему даже после изменений. С другой стороны, хоть какой-то характер движения мы выяснили. Более того, надеюсь, вам стало понятно, как применяются управляющие сигналы и как создаются диаграммы. А это и есть наша главная цель.

Создание более точной модели требует детального анализа системы, вывода уравнений ее состояния (что, в принципе, делается довольно просто), но самое главное и самое трудное – это собрать диаграмму из блоков. Вот вам «домашнее задание»: попробуйте самостоятельно получить уравнения и создать модель шарика в подвижной коробке, и пришлите нам ваши результаты по адресу: letters@linuxformat.ru. Небольшая подсказка: вы можете воспользоваться готовой демонстрационной диаграммой с именем **Bouncing_ball.cosf**.

Суперигра

Вернемся к нашим диаграммам. Я думаю, вы понимаете, что диаграммы не всегда можно уместить на одном листе: зачастую они бывают достаточно объемны, и их трудно отлаживать, а также обозревать. Некоторые блоки можно логически объединить в большие *суперблоки*, со своими входами и выходами. Для этой цели в *Scicos* существует специальный блок с именем SUPER_f. На **рис. 3** приведен пример «правильной» диаграммы для моделирования шарика в движущейся коробке, где части расчета координат *x* и *y* для шарика упакованы в суперблоки (мы-то уже решили эту задачку!).



› **Рис. 3. Главная диаграмма с суперблоком.**

Создать суперблок можно двумя способами: обычным, т.е. просто перенести в диаграмму и затем наполнить его содержимым, и путем преобразования уже имеющейся части диаграммы. Последнее выполняется следующим образом: выделяем несколько блоков, а затем последовательно выбираем пункты меню Diagram > Region to Super block.

Чтобы увидеть и изменить содержимое суперблока, достаточно щелкнуть дважды на его иконке. Открывшееся окно в системной строке-заголовке будет содержать текст **SuperBlock**, а кнопка Simulate станет недоступной, поскольку запускать модель на выполнение можно только из главного окна.

Каждый суперблок может содержать определенное число входных и выходных портов – как для данных, так и для управляющих сигналов. Таким образом, суперблок – это своего рода подпрограмма, и его можно отлаживать отдельно от основной части диаграммы, а затем вставить в нее уже готовеньким.

Входной порт данных определяется блоком IN_f (палитра Sources), а выходной – OUT_f (палитра Sinks). Для управляющих сигналов используются CLKINV_f и CLKOUTV_f с соответствующих палитр. Каждый входной или выходной порт суперблока должен иметь уникальный для своей группы номер, т.е. все входные порты данных нумеруются с 1, все выходные также нумеруются с 1, и т.д. Во время отладки суперблока в виде отдельной диаграммы входные и выходные блоки можно заменить некими тестовыми константами или зависимостями.

Scicos содержит также средства отладки, но программисты сочтут их весьма и весьма убогими. Выбрав последовательно пункты меню Simulate > Debug level, вы можете установить уровень режима отладки:

- **0** нет отладки;
- **1** краткая отладочная информация;
- **2** подробная отладочная информация;
- **3** молчаливая отладка, то есть режим отладки включен, но информация не выводится.

Вся отладочная информация отображается в окне *Scilab*, а не в редакторе *Scicos*, так что не убирайте его далеко. Кроме изменения режима отладки, вы можете перенести в суперблоки блок Debug с палитры Others. Он вызывается первым при входе в суперблок и может содержать скрипт на языке Scilab, позволяющий вам выполнить те или иные действия. По умолчанию в нем стоит команда **pause**, приостанавливающая процесс выполнения.

В заключение хочется вновь сказать, что пусть *Simulink* удобнее, имеет больше возможностей и широко известен, а потому для него существует обширная документация, **но** он очень дорого стоит. *Scicos* намного моложе и пока что неказист, и не может составить серьезную конкуренцию своему более именитому собрату. Однако я думаю, что со временем все изменится. Число членов консорциума *Scilab* растет все быстрее и быстрее, благодаря ужесточению законодательства в нашей стране все больше людей смотрят в сторону *Scilab/Scicos*, а для свободного ПО это автоматически приводит к увеличению числа разработчиков. Присоединяйтесь и вы! Помните, что даже если вы не программист и плохо разбираетесь в каких-то моментах, вы всегда можете внести посильную лепту путем перевода или написания документации, а ее катастрофически не хватает. LXF

Перечень палитр блоков

- Sources (источники) – блоки, представляющие собой источники данных или сигналов.
- Sinks (приемники) – блоки вывода или отображения данных.
- Linear (линейные) – базовые линейные операторы и подсистемы.
- Non_Linear (нелинейные) – нелинейные функции и подсистемы.
- Matrix (Матричные) – простые и сложные операции матричной алгебры.
- Integer (Целое) – различные операции для работы с целыми числами.
- Events (События) – блоки управления событиями в модели и синхронизации.
- Thresholds (Граница) – блоки для генерации событий при обращении входной величины в нуль.
- Others (Другие) – блоки различной направленности, от подключения программ на других языках программирования до решения уравнений в частных производных.
- Branching (Ветвление) – управление перемещением данных в ссылках.
- Electrical (Электричество) – базовые электрические блоки.
- ThermoHydraulics (Термодинамика) – основные элементы термодинамики.
- OldBlocks (Старые блоки) – старые версии блоков.
- DemoBlocks (Демо блоки) – блоки из демонстрационных примеров.

Через месяц Мы посмотрим, что нового может предложить нам Scilab 5.

Kino: добавим эффекты

ЧАСТЬ 2 Обработка отснятого видео – именно то, ради чего мы и привлекли к процессу компьютер. **Алексей Маслий** покажет, как расположить сцены в нужном порядке, добавить заставки и «вытянуть» звук.

Полагаю, за прошедший месяц вы достаточно освоились с подключением вашей видеокамеры и загрузкой материалов в *Kino*. Понятно, что копирование видеофрагментов на компьютер – не самоцель, а промежуточный этап обработки, поэтому сегодня мы займемся монтажом и наложением эффектов. Для работы с видео в *Kino* используются два режима – это вкладки Редактор и Срезка.

По умолчанию (если, разумеется, вы не меняли настройки *Kino*) каждая сцена, т.е. непрерывный фрагмент видеоматериала, начинающийся от нажатия на камере кнопки Запись и заканчивающийся нажатием на Стоп, записывается в отдельный файл; во всплывающих подсказках вы также можете встретить термин «план». На вкладке Редактор осуществляется глобальная манипуляция сценами, т.е. компоновка видеоматериала. Сцены можно вставлять, удалять и перемещать; можно просматривать их свойства, в число которых входит имя файла, дата съемки, длительность, положение в видеопотоке в файле и на оригинальном носителе (кассете): все это показано на **рис. 1**. Таким образом можно убирать кадры, отснятые при случайном включении камеры, неудачные моменты и т.п., группировать сцены не в порядке записи, а тематически (например, фильм об отпуске можно условно поделить на «достопримечательности» и «встречи с родственниками»), а также вставлять в материал заранее подготовленные видеоролики.

› **Рис. 1. Исходная точка: набор отснятых сцен или планов, как вам будет угодно.**

Все сцены размещаются в левой части окна в колонке Раскладка. Каждая из них обозначается пиктограммой с содержимым первого кадра и перечислением некоторых свойств. Чтобы выбрать сцену, просто щелкните по ней мышью. При этом в области отображения видео будет показан первый кадр сцены.

Специальной команды для удаления сцены нет, но ее можно банально вырезать (нажав Ctrl+X либо выбрав соответствующую опцию в меню или на панели задач), после чего она исчезнет из списка проекта. Однако файл на диске не будет удален, так что если данный материал вам абсолютно не нужен, запомните его имя, а затем уничтожьте в любимом файловом менеджере.

Для перемещения сцены необходимо выделить ее, нажать левую кнопку мыши и протащить на новое место; при этом синяя полоска будет отмечать положение, в которое будет вставлена сцена при отпускании кнопки. Альтернативный вариант: вырезать сцену, выбрать в раскладке «план», перед которым она должна находиться, и вставить ее. Вставку можно выполнить и из внешнего видео-файла, еще не включенного в проект. Для этого нужно выбрать сцену в раскладке и воспользоваться меню Файл > Вставить перед или Файл > Вставить после, а затем указать файл в появившемся диалоге. Аналогичным образом можно импортировать и целые проекты *Kino*, если потребуется слить их воедино. При необходимости вставить в проект не один, а несколько фрагментов, можно один раз выбрать пункт меню Файл > Вставить..., а затем перетаскивать видео в раскладку мышью. Ролики будут добавляться после первого импортированного файла.

Kino позволяет включать в проект не только **.dv**, но и файлы в других форматах: они будут преобразованы в родной для *Kino* **.dv** автоматически. И последнее: хотя пункт меню Файл > Открыть позволяет выбрать любой видеофайл, на самом деле он предназначен для открытия проекта. Поэтому, если вы укажете простой видеоролик, программа закроет текущий проект и создаст новый.

Ножницы и клей

Сцены можно объединять или резать на части. Необходимость первого действия может быть обусловлена несколькими обстоятельствами. Если один сюжет разбит на ряд сцен (например, вы решили пропустить длительное ожидание некоего события), то логично слить их в одну. Второй важный момент – спецэффекты: они создают отдельные файлы со сценами, содержащими только выбранный вами эффект перехода. Будет логично присоединить их к «родительской» сцене, да и вообще все сцены, связанные переходами, лучше объединить в одну, чтобы потом случайно не нарушить с таким трудом подобранный порядок следования.

Разрезание сцены на части может пригодиться при вставке в ее середину какого-нибудь спецэффекта. Другой частый случай – чистка видеоряда, т. е. удаление «проблемных» кадров.

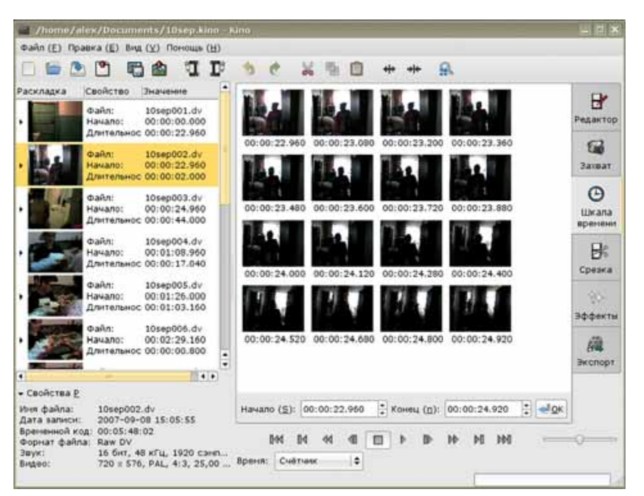
Объединение сцен выполняется по принципу «активная + следующая за ней». Иными словами, вы выбираете сцену (делаете ее активной), щелкаете пункт меню Правка > Склеить (или нажимаете кнопку Склеить в панели инструментов). В результате текущая и последующая сцены сливаются в одну.

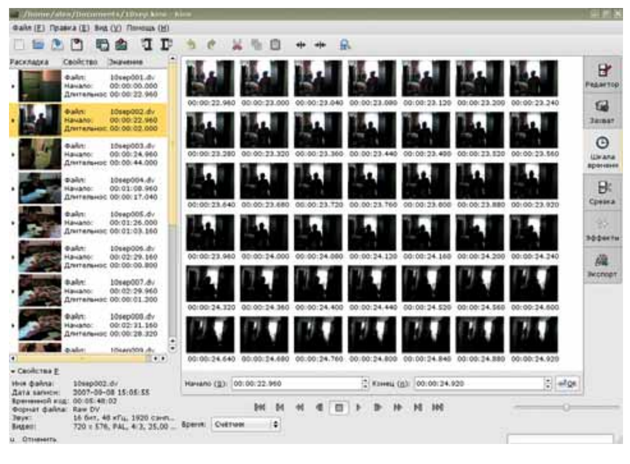
Для выполнения обратной операции, т.е. разрезания, необходимо выделить сцену, позиционироваться внутри нее на кадре, с которого должен начаться новый «план», и нажать в панели инструментов кнопку Разрезать или же воспользоваться меню Правка > Разрезать.

Позиционироваться внутри сцены можно тремя способами. Первый: активировать сцену, нажать кнопку Воспроизведение и в нужный момент нажать Стоп. Пожалуй, это наиболее неудобный вариант. Можно захватить мышью ползунок указателя положения в видеопотоке и, перемещая его, найти необходимый кадр. Однако и в этом случае добиться точного позиционирования будет нелегко, поскольку временная шкала относится не к отдельной сцене, а к проекту целиком. Здесь можно поступить так: открыть в новом проекте только одну сцену, и позиционироваться в ней (сначала – при помощи ползунка, а затем – используя кнопки покадрового сдвига).

И, наконец, третий способ – это использовать вкладку Шкала времени. Здесь вся сцена разбивается на временные интервалы с шагом,

» **Месяц назад** Мы подключили камеру и импортировали отснятый видеоматериал.





› Рис. 2 и 2а. «Цена деления» на шкале времени зависит от размера окна.

соответствующим количеству кадров, умещающихся в окне программы (сравните **рис. 2** и **2а**). Выбор конкретного кадра перемещает ползунок времени в соответствующую позицию. Если точка (кадр), по которой необходимо разрезать сцену, попала в один из временных промежутков, не беда. Можно детализировать картину, указав начало и конец отрезка, который вы хотите раскадрировать. Например, сцена длинной в 5 минут, начинающаяся в основном видеопотоке в позиции 00:10:30.000 (время указывается в режиме Счетчик), разбивается на 10 отрезков по 30 секунд. Допустим, вы обнаружили, что разрезать сцену надо где-то в шестом отрезке (т.е. в промежутке от 00:14:00.000 до 00:14:30.000). Введите два этих значения в поля Начало и Конец и нажмите кнопку OK. Выбранный временной промежуток снова разделится на 10 отрезков по 3 секунды. Если в такой раскладке опять не удастся выполнить точное позиционирование, то процесс можно продолжить с точностью до отдельного кадра.

Срезка? Срезка!

Режим Срезка для предназначен для редактирования отдельных сцен, т.е. вся работа происходит в рамках одного «плана» (**рис. 3**). Здесь производится монтаж видеоматериала более тонкими методами, чем доступны в Редакторе. У Срезки есть два режима работы: Перезапись и Вставка.

В режиме Перезапись выделенный фрагмент сцены заменяется другим, который берется из из **.dv**-файла. Для начала следует указать исходный фрагмент – по умолчанию выделена вся сцена. Точное положение начала и конца заменяемого фрагмента можно ввести вручную в специальных полях Начало и Конец (формат определяется режимом отображения). Фрагмент можно выделить и мышью, потянув за ползунки в виде прямоугольных треугольников под шкалой времени. Охваченная часть материала отрисовывается желтым цветом. Затем необходимо указать файл, которым будет заменен выделенный фрагмент и нажать кнопку Применить.



› Рис. 3. Режим «Срезка» в действии.

В режиме Вставка выделенный фрагмент появляется в виде отдельной сцены перед текущим «планом» или после него – это определяется кнопками Перед и После.

На мой взгляд, режим Перезапись является более востребованным. Основное его применение видится таким: сначала из сцены вырезается фрагмент, потом с ним производятся некие преобразования (например, повышается яркость или накладываются спецэффекты – кстати, все это может быть проделано и в другой программе), и, наконец, обработанный эпизод вставляется обратно в сцену на то же самое место.

Наложение спецэффектов

Kino предоставляет в ваше распоряжение базовый набор спецэффектов: создание заставок, обработку видео- и аудиопотоков, а также переходы. Первый из них создает отдельную сцену в виде набора кадров, второй применяется к уже имеющемуся «плану» или его фрагменту (но только в рамках одной сцены), а третий работает сразу с двумя сценами – текущей и последующей.

При выборе вкладки Эффекты центральное окно программы делится на четыре части: выбор режима работы и временного интервала, предпросмотр и эффекты для аудио- и видеопотоков. Последние, если они относятся к одному и тому же временному интервалу, можно накладывать параллельно.

Излишне говорить, что детально рассмотреть все эффекты в рамках одной статьи не получится, да и человеку творческому интереснее будет освоить их самостоятельно, методом проб и ошибок. Ограничимся лишь теми вариантами, которые наверняка пригодятся вам в обработке домашнего видео.

Минуточку внимания!

Прежде чем начать, обратите внимание на раздел Вывод в левой верхней части окна эффектов. Здесь указывается каталог и префикс имен файлов, в которых будет сохраняться созданное видео. Не рекомендую принимать установки по умолчанию, т.к. в этом случае файлы будут создаваться непосредственно в вашем домашнем каталоге, и если вы работаете над несколькими проектами, разобраться, где что, будет довольно проблематично.

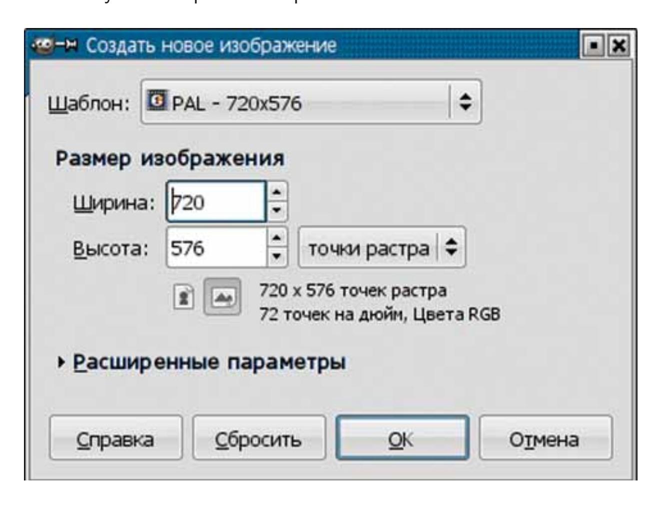
Заставки, заставки, заставки!

Как мне кажется, статические заставки разумно помещать в начале каждого большого (и не очень большого) сюжета. Их содержимое – в

большой степени дело вкуса. Лично я обязательно выношу на заставку дату съемки и поясняющую информацию («Первые шаги», «С друзьями на лыжах» и т.д.) В дальнейшем, при просмотре фильма, это помогает лучше ориентироваться в происходящем.

Заставки можно создавать в любом графическом редакторе, позволяющем сохранять файлы в формате **.png**. Я предпочитаю использовать *GIMP*, хотя бы потому, что он всегда под рукой, какой бы дистрибутив вы ни выбрали.

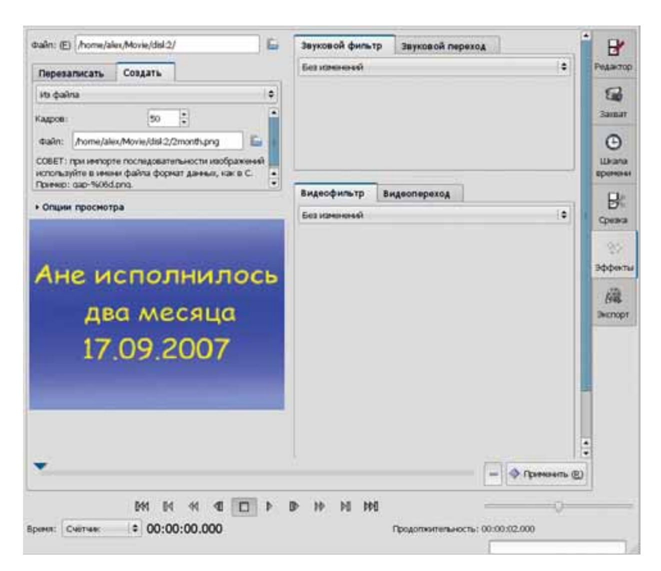
Процессу создания изображений-шедевров в *GIMP* посвящена целая серия учебников **LXF**, поэтому я опять буду краток. Запустите редактор и выберите в меню Файл > Создать. В появившемся окне, Создать новое изображение, воспользуйтесь выпадающим списком Шаблон, и укажите в нем PAL или NTSC, в зависимости от формата, в котором выполнена запись (**рис. 4**). Нажмите OK, чтобы создать новый рисунок. Далее все зависит от вашей фантазии: используйте простую или градиентную заливку в качестве фона, нарисуйте что-нибудь или воспользуйтесь сторонними наработками.



› Рис. 4. Создаем заставку в *GIMP*.

В качестве заставки также можно взять и кадры из фильма, полученные при помощи функции Стопкадр или экспорта изображений на вкладке Экспорт. Результирующую картинку следует открыть в *GIMP*, чтобы наложить надписи. Рекомендую выбирать кадры, на которых есть большие однотонные пространства (небо, море) или сильно затененные области: на этом фоне надписи будут легко читаемы.

Закончив работу над заставкой в *GIMP*, переключитесь в *Kino*. Вообще-то все равно, где именно создавать сцену с заставкой, поскольку ее всегда можно переместить в нужную позицию, но я предпочитаю сразу расставлять все на свои места. Если вы солидарны со мной в данном вопросе, выберите в раскладке сцену, перед которой надо вставить заставку и перейдите на вкладку Эффекты. Откройте вкладку Создать, в выпадающем меню укажите Из файла, введите количество кадров, из которых будет состоять сцена (чаще всего камеры снимают 25 кадров в секунду, так что данное число можно легко перевести во время) и, наконец, выберите файл. Он сразу же отобразится в окне предпросмотра (**рис. 5**). Если больше никаких эффектов накладывать не планируется, можно нажимать кнопку Применить – будет создана новая сцена. Я обычно создаю две одинаковые сцены длительностью по 2 секунды (50 кадров): вторая используется затем для создания видеоперехода.



› Рис. 5. Окно предпросмотра с нашей заставкой.

Эффектный звук

Начну с грозного предупреждения: нажатие кнопки Применить безвозвратно портит как аудио-, так и видеопоток (производится перезапись исходного **.dv**-файла). Так происходит потому, что **.dv**-файлы обычно весьма объемны, и создание резервных копий способно забить под завязку диск практически любого размера. Поэтому всегда проверяйте соответствие желаемого и действительно результата в окне предпросмотра, и только потом накладывайте эффект на материал.

Для обработки аудиопотока в *Kino* предусмотрено два режима: Звуковой фильтр и Звуковой переход. Первый позволяет выполнять базовые преобразования звуковой дорожки. По умолчанию выбран фильтр Без изменений; список возможных вариантов меняется от версии к версии (а возможно, и от сборки к сборке), однако в любом случае ассортимент будет существенно меньше, чем в специализированном звуковом редакторе. Поэтому для сложной обработки аудио я применяю *Audacity*: мы писали о нем в LXF98 и LXF106.

Но все же, коль скоро такая опция в программе есть, попробуем разобраться, как ею пользоваться. Во-первых, необходимо переключиться в режим Перезапись. Далее, активируйте в раскладке сцену, на которую будет накладываться эффект: если это фрагмент, следует указать временной диапазон и выбрать фильтр из выпадающего списка. Еще раз настоятельно рекомендую предварительно прослушать, что получилось, нажав Play. Если вам понравился результат, нажмите кнопку Применить.

В режиме Звукового фильтра доступны следующие функции:

» Дублирование. Позволяет наложить новую звуковую дорожку вместо уже имеющейся. Это пригодится, когда вы обработаете аудио-поток во внешнем редакторе, а потом решите вернуть его на место, в сцену.

» Смешивание. Микширует две звуковые дорожки, например, добавляет фоновую музыку. Смешивание, как и Дублирование, требует наличия внешнего аудиофайла. Можно также указать смещение аудио-потока относительно видеоряда. Эти два фильтра, пожалуй, являются наиболее полезными среди всех средств обработки звука, имеющихся в *Kino*.

» Нарастание, Убывание и Усиление работают с текущей звуковой дорожкой сцены. После задания временного отрезка, первый из них увеличивает громкость звука с нуля до текущего уровня; второй, наоборот, уменьшает с текущего уровня громкости до нуля. Третий позволяет провести примитивную обработку аудиосигнала, т.е. усилить или ослабить звук.

» Тишина. Как следует из названия, данный фильтр убирает звук в выбранной сцене или ее отрезке.

Звуковой переход – это режим, автоматически создающий переход аудиопотока от одной сцены к другой. Выпадающее меню содержит всего два пункта: Без изменений и Наплыв, т.е. фактически вам доступен один эффект: затухание звука на первой сцене и его плавное нарастание на второй. Имеются также две кривые уровня громкости (для двух сцен), которые можно изгибать, чтобы добиться желаемого результата. Такой переход часто можно услышать в аудиозаписях, а в проигрывателях данный эффект обычно называется «Crossfade».

Я обещал вам рассказать о видеопереходах, но пока что коснулся только статических заставок. Не думайте, что возможности *Kino* ими ограничиваются – все самое интересное традиционно припасено на следующий раз! **LXF**

» **Через месяц** Финальная стадия: создадим видеопереходы и экспортируем фильм на DVD.

Финальные штрихи

ЧАСТЬ 4 Уроки *Django* подходят к концу, наступает пора экзаменов – если не для вас, то для приложений уж точно. **Никита Шультайс** разберется с тестированием и пробежится по другим возможностям этого каркаса.



Наш эксперт

Никита Шультайс
Узнал о *Django* два года назад на LUG'e и с тех пор использует его как базовый инструмент для разработки web-приложений.

Новостная лента готова – но оправдает ли она ожидания пользователей? Ответ на этот вопрос может дать тестирование, лучше всего – в реальных условиях; но какие-то основные вещи можно проверить еще на этапе разработки. Если вы успели заглянуть в учебник Rails (стр. 66), то уже знаете о методологии TDD. Мы, однако, пойдем другим путем, и будем тестировать приложение не до написания, а после.

Дело это – серьезное и сложное, так как нам нужно учитывать взаимодействие с БД, компиляцию шаблонов, обработку GET- и POST-запросов и прочие компоненты системы: сбой в любом из них может вызвать нарушить работу всего сайта. К данной задаче можно подойти с двух сторон:

» Тестирование в браузере. Речь идет о программах *Twill* (**twill.idyll.org**) и *Selenium* (**selenium.openqa.org**): они «запоминают» последовательность ваших действий для каждой страницы, а затем воспроизводят ее по запросу. Например, можно ввести в поля формы заведомо неверные данные, получить ожидаемую ошибку и повторять этот тест при каждом серьезном изменении в коде вашего приложения.

» Тестирование на сервере. И тут *Django* не оставляет нас на произвол судьбы, предлагая сразу два варианта: *doctest* (тестирование через документацию) и *unittest* (модульное тестирование), плюс специальный клиент для отправки GET- и POST-запросов.

Если вы давно программируете на Python, то вам, наверное, будет ближе *doctest*, а мигрантам из мира Java скорее придется по вкусу *unittest*. Никаких ограничений на их использование не накладывается: вы можете выбрать одну систему или применять обе сразу. Мы же остановимся на *doctest*.

Документируй это!

Строка документации в Python – это обычный текст, размещаемый после определения функции или класса непосредственно в исходном коде. Она же предоставляет содержимое для атрибута __doc__. Как правило, ее помещают в тройные кавычки ("""), что позволяет вводить сложные конструкции с переносами строк, отступами, теми же кавычками и... тестами. Этим мы и воспользуемся.

Тесты могут находится в файлах моделей (**models.py**) – для проверки последних – и в специальных файлах **tests.py**, расположенных в директории приложения. К примеру, создайте файл **news/tests.py** такого содержания:

```
1  # -*- coding: utf-8 -*-
2
3  """
4  >>> from news.models import News
5  >>> from datetime import datetime
6  >>> news = News(title="Заголовок",description="Описание",pub_date=datetime.now(), text="Текст")
7  >>> news.save()
8  >>> news = News.objects.get(pk=1)
9  >>> print news.title.encode('utf-8')
10 Заголовок
11 """
```

В строке 1 задается кодировка, а начиная с третьей идет сам тест. Заметьте, что каждая строка начинается с трех знаков «больше» (>>>), как в интерактивном режиме работы Python. В строке 10 этих знаков нет, так как она содержит ожидаемый вывод команды **print** со строки 9.

Перед тем, как запустить тест, в **settings.py** нужно добавить строку

```
TEST_DATABASE_CHARSET="UTF8"
```

чтобы кодировки файла и базы данных совпадали. Перед выполнением теста *Django* создаст специальную вспомогательную БД, поэтому пользователь, указанный в **settings.DATABASE_USER**, должен иметь соответствующие права. Для начала тестирования введите команду:

```
python manage.py test news
```

после выполнения которой вы увидите примерно то, что показано на **рис. 1**.

Сообщения похожи на появляющиеся во время создания таблиц при первой установке, но теперь все происходит в тестовой БД. В конце отображается число выполненных тестов, их результаты и уведомление об уничтожении тестовой базы. Мы проверяли наше приложение (*news*), но, как вы помните, *Django* содержит несколько собственных приложений и представлений (например, «админку») – и они тоже снабжаются своими тестами. Чтобы выполнить их все, нужно ввести команду:

Месяц назад Интернационализация и кэширование: без них не обходится ни один настоящий сайт.

```
Creating test database...
Creating table auth_message
Creating table auth_group
Creating table auth_user
Creating table auth_permission
Creating table django_content_type
Creating table django_session
Creating table django_site
Creating table django_admin_log
Creating table news_comment
Creating table news_news
Installing index for auth.Message model
Installing index for auth.Permission model
Installing index for admin.LogEntry model
Installing index for news.Comment model
.
----------------------------------------
Ran 1 test in 0.008s

OK
Destroying test database...
```

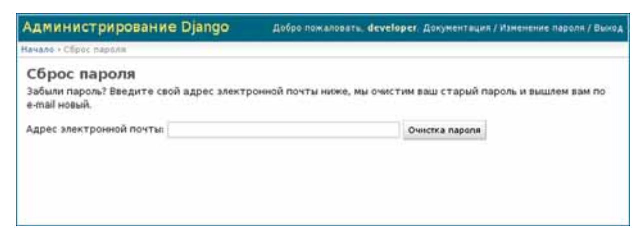
› Рис. 1. Тест пройден удачно!

```
python manage.py test
```

добавив предварительно в главный файл URL-карт следующие строки:

```
urlpatterns += patterns('django.contrib.auth.views',
   url(r'^auth/password_reset/$','password_reset'),
)
```

Здесь мы подключаем одно из встроенных представлений, предназначенное для восстановления пароля. Это необходимо, так как при тестировании всего проекта происходит обращение по указанному URL, и если представление не определено, тест будет провален. Кстати, вы тоже можете попробовать password_reset в работе (**рис. 2**).



› Рис. 2. Забыли пароль? Это проблемы Django!

Имитатор Сети

Количество тестов уже достигло шести, но помимо создания и извлечения объектов из базы, нам нужно проверить реакцию на GET- и POST-запросы. Как вы знаете, для этих целей существует специальный клиент: он эмулирует запрос и возвращает переменные, которые передаются в шаблон для данного URL. Добавьте в файл **tests.py** после строки 10 следующий код:

```
11 >>> from django.contrib.auth.models import User
12 >>> user = User.objects.create_user('django_guru', 'user@example.com', 'password')
13 >>> user.save()
14 >>> from django.test.client import Client
15 >>> c = Client()
16 >>> c.login(username='django_guru',password="password")
17 True
18 >>> response = c.get('/news/1/')
19 >>> response.status_code
20 200
21 >>> print response.context[0]['user'].username
22 django_guru
23 >>> response = c.post('/news/1/',{'username':"testuser",'text':»»})
24 >>> response.status_code
25 200
26 >>> response = c.post('/news/1/',{'username':"testuser",'text':»Comment text»})
27 >>> response.status_code
28 302
29 >>> from news.models import Comment
30 >>> comment = Comment.objects.get(news=news)
31 >>> print comment.text
32 Comment text
```

Разберемся, что здесь происходит. В строках 11–13 мы создаем нового пользователя (django_guru), а в 14–15 – тестовый клиент. В строке 16 django_guru авторизуется, и отныне все действия в системе будут совершаться от его имени. В строке 18 мы переходим на страницу нашей первой новости, передав средствами клиента GET-запрос. Для проверки, что нам это удалось, мы изучаем код ответа сервера (строка 19) – он должен равняться 200, или тест будет провален. Затем (строки 21–22), чтением дополнительных данных ответа, мы убеждаемся, что запрос сделал зарегистрированный пользователь django_guru. Теперь самое время оставить комментарий – не зря же мы авторизовались? В строке 23 генерируется POST-запрос (второй аргумент метода post() – словарь отсылаемых на сервер данных). Обратите внимание, что значение ключа text оставлено пустым, а значит, комментарий не добавится, однако сервер по-прежнему должен возвращать код 200 (строка 25). А вот в строке 26 мы передаем все необходимые данные, и так как после создания комментария нас перенаправляют на страницу новости, код ответа должен быть равен 302 (Требуемый URL перемещен). В строках 29–32 проверяется, что комментарий был действительно добавлен: мы сравниваем его текст с исходным значением. Уфф... тест пройден.

Действительно простая синдикация

Какой новостной сайт без ленты? RSS и/или Atom есть везде – будут и у нас, а *Django* нам в этом поможет. Откройте главный файл URL-карт и добавьте в его конец следующие строки:

```
1  from feeds import LatestNews
2
3  feeds = {
4     'latest': LatestNews,
5  }
6  urlpatterns += patterns('',
7     (r'^feeds/(?P<url>.*)/$', 'django.contrib.syndication.views.feed',
8        {'feed_dict': feeds}),
9  )
```

Далее нужно приготовить ленту LatestNews, которую мы импортируем в строке 1. Создайте в корне проекта каталог **feeds** с файлом **__init__.py** следующего содержания:

```
# -*- coding: utf-8 -*-

from django.contrib.syndication.feeds import Feed
from news.models import News
from django.contrib.syndication.feeds import FeedDoesNotExist

class LatestNews(Feed):
   title = "Последние новости нашего сайта"
   description = "Последние события на сайте mysite.com"
   link = "http://127.0.0.1:8000/news/"

   def items(self):
      return News.objects.order_by("-pub_date")[:5]

   def item_link(self, obj):
      from django.core.urlresolvers import reverse
      return 'http://127.0.0.1:8000%s' % reverse('news.news_detail',kwargs={"news_id":obj.pk})
```

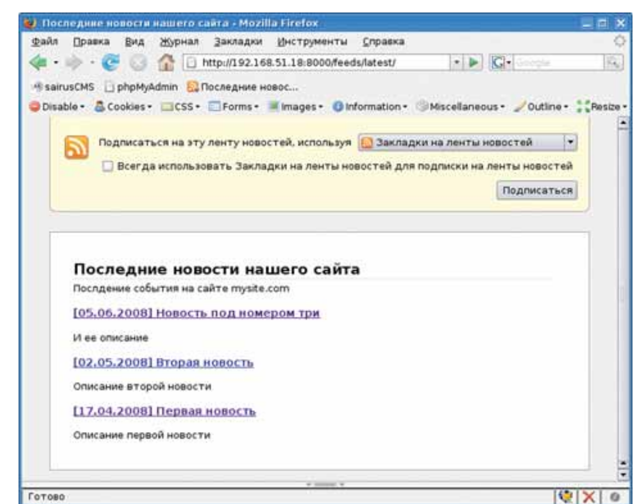
Поля title, description и link класса LatestNews являются обязательными и отвечают за одноименные элементы RSS. Метод items() передает в ленту требуемые объекты, а item_link() отвечает за ссылку на сайт. Теперь создайте каталог **feeds** в **media/templates** и добавьте в него два файла, **latest_description.html** и **latest_title.html**: они будут отвечать за вид новостной ленты. В **lates_description.html** напишите:

```
{{ obj.description }}
```

а в **latest_title.html**:

```
[{{ obj.pub_date|date:"d.m.Y" }}] {{ obj.title }}
```

Объект obj представляет собой запись из выборки, которую мы возвращаем в строке 13 файла **feeds/__init__.py**. Пройдя по адресу http://127.0.0.1:8000/feeds/latest/, мы увидим предложение *Firefox* сохранить ленту новостей. Пользователи KDE, вероятно, предпочтут *Akregator* – с ним тоже нет никаких проблем (**рис. 3, 4**)



› **Рис. 3. Firefox предлагает подписаться на обновления нашего новостного сайта – теперь держитесь!**

Общие представления

Чтобы облегчить жизнь web-разработчика, *Django* включил большое число представлений для решения стандартных задач. Так, добавив в главный файл URL-карт следующий код:

```
from django.views.generic.list_detail import object_list
from news.models import News
urlpatterns += patterns('',
    ('^lastnews/$', object_list, {
        'queryset': News.objects.all().order_by('-pub_date')[:10],
        'template_name':'news/last_news.html',
        'template_object_name':'last_news'})
)
```

а также заменив в файле **news/templates/news/last_news.html**

```
{% for news in last_news %}
```

на

```
{% for news in last_news_list %}
```

мы сможем просматривать последние новости по адресу http://127.0.0.1:8000/lastnews/, не вызывая представление news.last_news. Чтобы сделать доступными оба варианта, нужно найти в представлении news.last_news строку

```
"last_news":news,
```

и заменить ее на

```
"last_news_list":news,
```

Как вы уже догадались, общее представление object_list предназначено для работы со списком объектов. Еще есть представления для вывода объектов в зависимости от даты (django.views.generic.date_based.*), что позволяет очень просто создавать архивы записей:

» archive_index – вывод последних объектов, добавленных в базу данных;
» archive_{year,month,week,day,today} – вывод всех объектов за определенный год, месяц, неделю, день или за сегодня;
» object_detail – вывод одного объекта за определенный день.

Доступны общие представления для создания, обновления и удаления объектов. Все они работают немного быстрее, чем созданные вручную, но позволяют решать лишь самые элементарные задачи. Если данные в вашем приложении выбираются из нескольких таблиц и это сопровождается расчетами, то общие представления не помогут – на то они и общие.

Добавляем переменные на лету

В глубинах *Django* скрываются глобальные контекстные процессоры, основная задача которых – снабжать шаблоны переменными и объектами. Узнать, какие из них подключены, можно в кортеже TEMPLATE_CONTEXT_PROCESSORS в файле **settings.py**. Например, у нас сейчас работают следующие процессоры:

» auth – информация о пользователе: объект user, его права доступа и сообщения, которые были ему отправлены;
» i18n – сведения о текущем языке сайта и клиента;
» request – данные о запросе.

Кроме них, существует еще процессор debug, передающий в шаблон данные о выполненных SQL-запросах, плюс мы можем написать свой собственный! Для этого создадим в корне нашего проекта каталог processors и добавим в него два файла: **__init__.py** и **context_processors.py**. Последний должен содержать такой код:

```
import settings
  def site_settings(request):
    return {'SETTINGS': settings}
```

Чтобы подключить процессор, просто перечислите его в кортеже TEMPLATE_CONTEXT_PROCESSORS. Проверим работоспособность: добавим в шаблон **news.html** следующее:

```
{{ SETTINGS.TIME_ZONE }}
```

Конечно, TIME_ZONE можно заменить на любую другую переменную, определенную в **settings.py**.

Сам себе фильтр

С фильтрами мы познакомились еще в LXF105, однако часто возникают ситуации, когда поставляемых с *Django* вариантов недостаточно. Чтобы написать свой собственный фильтр, создайте в корне проекта каталог **templatetags/** и добавьте в него файлы **__init__.py** и **filters.py**. В **filters.py** напишите:

```
1  from django import template
2
3  register = template.Library()
4
5  @register.filter
6  def exp(value, arg):
7      if value.isdigit() and arg.isdigit():
8          return int(value)**int(arg)
9      else:
10         return '<span style="color:red">Error</span>'
11 exp.is_safe = True
```

Мы создали фильтр exp, который получает значение и показатель степени и возводит одно в другое; если аргументы не являются числами, генерируется ошибка. В строке 5 мы регистрируем фильтр в системе с помощью декоратора. Строка 11 указывает, что exp может возвращать HTML-код. Поскольку (в целях безопасности) он автоматически экранируется (< и > заменяются на < и > и т.д.), то, желая видеть чистый HTML, мы должны запретить такое поведение вручную. Следующим шагом является подгрузка библиотеки фильтров в шаблон, для чего нужно добавить в него следующую строку:

```
{% load filters %}
```

Вообще-то *Django* ищет созданные библиотеки шаблонов в корне приложения, поэтому наш фильтр пока не будет доступен. Это не очень удобно, особенно если мы хотим использовать один и тот же фильтр во многих приложениях. Решение – создать для проекта единую библиотеку, а в приложения помещать лишь символьные ссылки на нее.

```
ln -s /var/www/myproject/templatetags/ /var/www/myproject/news/
```

Теперь проверим работу фильтра, добавив в какой-нибудь шаблон строку.

```
{{ "4"|exp:"4" }}
```

Во время компиляции она будет заменена на 256. Если же мы напишем

```
{{ "a"|exp:"4" }}
```

то увидим слово «Error», выделенное красным цветом.

Кстати, если бы мы не указали в строке 11 фильтра exp.is_safe = True, можно было бы просто применить фильтр safe прямо в шаблоне:

```
{{ "a"|exp:"4"|safe }}
```

После регистрации фильтра в системе, информация о нем становится доступной по адресу http://127.0.0.1:8000/admin/doc/filters/ (**рис. 4**)

filters
To use these filters, put {% load filters %} in your template before using the filter.
exp

› Рис. 4. Система любезно подскажет, как использовать созданный вами фильтр.

Компоненты

Если нам надо выполнить какие-либо действия до или после того, как будет вызвано представление, либо в случае ошибки, можно создать свой компонент (middleware) или воспользоваться поставляемым с *Django*. Мы уже делали это, когда изучали кэширование (LXF107). Напомню, что в файле **settings.py** есть кортеж MIDDLEWARE_CLASSES, который перечисляет компоненты, задействованные в проекте. У нас это:

» django.middleware.common.CommonMiddleware Решает общие задачи: нормализует URL (добавляет префикс www и завершающий /), запрещает доступ к сайту определенным роботам, взаимодействует с Etag.

» django.contrib.sessions.middleware.SessionMiddleware Это сессии.

» django.contrib.auth.middleware.AuthenticationMiddleware А это – авторизация.

» django.middleware.doc.XViewMiddleware Используется для автоматического документирования *Django*.

» django.middleware.locale.LocaleMiddleware Интернационализация.

Помимо перечисленных выше, в *Django* доступны следующие компоненты (django.middleware.*):

» gzip.GZipMiddleware Сжатие отправляемой страницы для экономии трафика.

» http.ConditionalGetMiddleware Поддержка условных GET-запросов для работы с Last-Modified и Etag.

» http.SetRemoteAddrFromForwardedFor Обратное проксирование.

» cache.CacheMiddleware Тот самый кэш, с которым мы сталкивались на прошлом уроке.

» transaction.TransactionMiddleware Компонент для включения в SQL-запросы транзакционных конструкций: COMMIT, ROLLBACK. Заметьте, что не все базы данных поддерживают транзакции.

И, наконец, django.contrib.csrf.middleware.CsrfMiddleware, защищающее от CSRF-атак.

По сложившейся уже традиции, рассмотрим, как написать свой собственный компонент. С точки зрения программиста, это просто класс Python, имеющий ряд методов, вызываемых *Django* в определенные моменты времени. Первым из них является конструктор __init__(self), регистрирующий компонент в системе. Дальше следуют методы, определяющие порядок выполнения кода:

» process_request() – запускается после того, как поступил запрос, но перед тем как *Django* начнет искать запрашиваемый адрес в URL-картах;

» process_view() – отрабатывает, когда конкретное представление уже определено, но еще не запущено;

» process_response() – выполняется после представления. Используется для сжатия сгенерированного HTML.

» process_exception() – вызывается, если что-то пошло не так или было возбуждено необработанное исключение.

Вот, в сущности, и все. Впрочем, нет – взгляните на врезку *И прочая, прочая, прочая*, почитайте документацию или свободную книгу о *Django* – Django Book (www.djangobook.com); если же вам больше по душе русский, советую заглянуть на cargo.caml.ru/djangobook. Наконец, примените полученные знания на практике – и дайте нам знать, если у вас получится что-то действительно стоящее! LXF

И прочая, прочая, прочая...

За четыре урока мы успели рассмотреть почти все возможности *Django*, но кое-что осталось неохваченным...

» **Электронная почта**

Django предлагает высокоуровневый API для отправки письма в одно действие:

```
from django.core.mail import send_mail
send_mail('Тема', 'сообщение.', 'from@example.com', ['to@example.com'], fail_silently=False)
```

Кроме того, есть функции массовой рассылки сообщений, оповещения администраторов и менеджеров сайта, а также работы с различным содержимым (HTML, текст, графика и т.п.)

» **CSV и PDF**

Django позволяет легко формировать файлы с данными, разделенными запятыми (CSV), а также PDF-документы, используя библиотеку *ReportLab* (www.reportlab.org/rl_toolkit.html).

» **Постраничный вывод**

Когда количество объектов настолько велико, что одной страницы становится мало, на помощь приходит специальный класс Paginator, который помогает организовать постраничный вывод:

```
>>> from django.core.paginator import Paginator
>>> objects = ['django', 'python', 'mysql', 'apache']
>>> p = Paginator(objects, 2)
>>> page1 = p.page(1)
>>> page1.object_list
['django', 'python']
```

Поскольку большинство наших объектов хранятся в базе данных, *Django* также предлагает класс QuerySetPaginator, который принимает не список, а множество объектов из СУБД.

» **Карты сайта**

Хотите, чтобы ваш сайт правильно индексировался поисковыми машинами? Необходимо создать его карту! *Django* поможет и здесь, а функция django.contrib.sitemaps.ping_google «заставит» Google обновить индексы для вашего сайта.

» **Управление несколькими сайтами**

Одной из задач, с которой успешно справляется *Django*, является управление несколькими схожими по тематике сайтами из одной инсталляции. Проект первоначально разрабатывался как платформа для новостных порталов, и одна новость могла появиться сразу на нескольких ресурсах.

» **В помощь дизайнерам**

Подгрузив к шаблонам модуль помощи дизайнерам:

```
{% load webdesign %}
```

вы получите в свое распоряжение тэг {% lorem %}, с помощью которого можно выводить известную латинскую фразу «lorem ipsum ...» для заполнения шаблонов «рыбой».

» **И более того**

Django содержит много других «вкусностей», которые очень хорошо описаны в прилагающейся документации на английском языке.





Утечка ресурсов – в первую очередь, памяти – одна из проблем современных сложных приложений. **Андрей Кузьменко** покажет, как избежать ее в ваших программах.



Наш эксперт

Андрей Кузьменко
Убежденный сторонник надежного ПО и любитель C++. Из всех дистрибутивов Linux отдает предпочтение Knoppix.

Можно смело считать 2007 год переломным моментом в переходе на многоядерные процессоры и активном внедрении многопоточного программирования в повседневную практику. Компании Intel и AMD обозначили производство многоядерных процессоров как основное направление своего развития на ближайшие несколько лет. Покупая современный ноутбук или ПК, вы обязательно увидите шильдик Core2Duo или Athlon X2. В текущем году производители чипов уже делают упор на четырехъядерные процессоры. Очевидно, что производители ПО должны адекватно реагировать на эти изменения, выпуская продукты, задействующие все преимущества новых технологий. А на чем они пишутся?

В языке C++, в отличие, например, от Java, отсутствует встроенная поддержка многопоточного программирования. Иными словами, с помощью «голого» C++ нельзя создавать соответствующие стандарту языка многопоточные приложения. Разумеется, данная функциональность все же доступна в виде библиотек, например, *Pthreads* (POSIX Threads), особенной популярной в мире Unix. Кроме нее, разработчик может использовать библиотеку *Boost* (www.boost.org) или *ThreadWeaver* (api.kde.org). Коммерческие Unix-системы, например, Sun Solaris, предлагают собственные библиотеки многопоточного программирования.

На данном уроке мы рассмотрим некоторые проблемы, возникающие при написании программ на C++ с использованием библиотеки *Pthreads*. Наряду со встроенными базовыми типами (int, char, double) в функциях работы с потоками могут использоваться объекты классов. Вот тут-то нас и подстерегают проблемы и неожиданности, которые мы сегодня обсудим. Кроме этого, мы оценим работу многопоточных программ, написанных на C++ и *Pthreads*, в различных Linux-системах.

И снова классы…

Одной из серьезных проблем языка C++ (настолько серьезной, что иные компании создают целые платформы со сборкой мусора, лишь бы с нею не сталкиваться) является утечка ресурсов. Это может быть память, выделенная с помощью оператора new, файловые дескрипторы, сетевые сокеты, мьютексы. Для освобождения ресурсов, используемых объектом, в C++ предусмотрены специальные методы – деструкторы. Задача программиста – правильно написать деструктор и убедиться в том, что в программе происходит его вызов. Только так можно гарантировать, что ресурсы будут возвращены системе.

Во всех тестах, описываемых в этой статье, будет использоваться класс checker, объявленный в файле **checker.hpp**. Вот он:

```
class checker{
  private: string name;
  public:
  explicit checker(string s) ;
  ~checker( );
  void calc(long N);
  void say_hello(void);
};
```

Класс обладает «говорящими» конструктором и деструктором, метод calc() имитирует длительную по времени расчетную задачу, а функция say_hello() выводит сообщение на консоль.

С вещами на выход!

Выполнение потоковой функции может быть прервано по трем причинам:

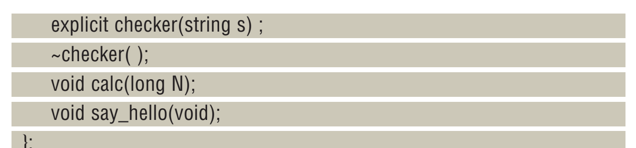
1. В результате «естественного завершения» оператором **return**;
2. В результате вызова функции pthread_exit();
3. В результате аннулирования другим потоком.

Особый интерес для нас будет представлять вызов деструкторов объектов в случаях 2 и 3.

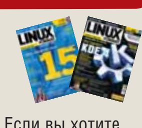
В документе The Open Group Base Specifications Issue 6 IEEE Std 1003.1, 2004 Edition в разделе, посвященном функции pthread_exit(), по этому поводу говорится следующее [здесь и далее перевод автора]: *«Функция void pthread_exit(void *value_ptr) завершает вызывающий поток и делает значение value_ptr доступным для успешного присоединения к завершающему потоку. Любые обработчики отмены, помещенные в стек, но еще не извлеченные из него, будут извлечены в порядке, обратном по отношению к порядку помещения в стек, а после выполнены. Если потоку принадлежат данные, то после выполнения всех обработчиков отмены будут вызваны соответствующие функции деструкторов, при этом порядок их вызова не определен. При завершении потока ресурсы процесса, включая мьютексы и дескрипторы файлов, не освобождаются, и не выполняются никакие восстановительные действия уровня процесса, включая всевозможные вызовы любых функций atexit()».* То есть, если объект в потоковой функции представляет собой локальную переменную с классом памяти auto, то после вызова pthread_exit() для него должен быть автоматически выполнен деструктор. Однако, как показывает практика, бывают случаи, когда этого не происходит.

Рассмотрим следующую программу:

```
void* task1(void *X) {
  std::cout<<" Start task_1!"<<std::endl;
  checker P("First"); P.calc(5);
  pthread_exit(NULL);
  return 0;
}
void* task2(void *X) {
  std::cout<<" Start task_2!"<<std::endl;
  checker Q("Second");  Q.calc(8);
```



Мы про это уже писали



Если вы хотите узнать больше о программировании с использованием *Pthreads*, обратитесь к нашим учебникам, опубликованным в LXF86-87/88.

в паутине

```
  return 0;
}
int main(void){
  std::cout<<" Start test #1!"<<std::endl;
  pthread_t threadA, threadB;
  pthread_create(&threadA, NULL, task1, NULL);
  pthread_detach(threadA);
  pthread_create(&threadB, NULL, task2, NULL);
  pthread_detach(threadB);
  pthread_exit(NULL);
  return 0;
}
```

Здесь объявлены две потоковых функции: task1() и task2(). В качестве элемента данных в каждой из них используется объект класса checker, выделенный в стеке. Функция task1() завершается принудительно с помощью pthread_exit(), а task2() выходит «естественным образом» через return 0. При этом потоки создаются как открепленные, т.е. при их уничтожении ресурсы, которые они использовали, сразу же возвращаются системе.

Рассмотрим результат работы программы в различных дистрибутивах Linux. Например, в SLAX 6.0.3 вывод на консоль будет таким:

```
Start test #1!
Start task_1!
Constructor done! Name:First
Start task_2!
Constructor done! Name:Second
Destructor done! Name:First
Destructor done! Name:Second
```

А вот что получается в ALT Linux 3.0.4:

```
Start test #1!
Start task_1!
Constructor done! Name:First
Start task_2!
Constructor done! Name:Second
Destructor done! Name:Second
```

Видите? Деструктор объекта из потоковой функции task1() вызван не был! Конечно, 3.0.4 – не самая актуальная версия данного дистрибутива, но, как мы увидим далее, аналогичные проблемы имеют место и в более современных ОС.

Поел – убери за собой

Очень часто при завершении потока бывает необходимо выполнить некоторые заключительные операции: освободить память, закрыть файлы, снять блокировки с разделяемых переменных и т.п. Желательно, чтобы эти действия выполнялись единообразно, как для стандартного завершения потоковой функции оператором return, так и при аннулировании другим потоком. Библиотека *Pthreads* предоставляет для этого возможность, называемую «стеком очистительно-восстановительных операций». Как она работает? С каждым потоком, имеющимся в программе, связывается стек очистительно-восстановительных операций, который содержит указатели на функции, вызываемые во время аннулирования (завершения) потока. Для работы с данным стеком используются две функции (или макроса):

» pthread_cleanup_push() Принимает в качестве параметров указатель на помещаемую в стек функцию и передаваемый ей аргумент;

» pthread_cleanup_pop() Принимает в качестве параметра целочисленное значение и извлекает завершающую функцию с вершины стека. Если аргумент отличен от нуля, завершающая функция выполняется.

Давайте рассмотрим еще один пример. Здесь мы определяем потоковую функцию, которая в зависимости от значения параметра, заданного пользователем, завершается либо «обычным образом», либо вызовом pthread_exit():

```
void* task1(void *X){
  std::cout<<" Start test thread!"<<std::endl;
  checker *Z = new checker("agent");
  pthread_cleanup_push(del_ptr_checker, Z);
  int *counter = static_cast<int*>(X);
  for(int i=0; i<(*counter); ++i)
  {
    if(i==1000) pthread_exit(NULL);
  }
  pthread_cleanup_pop(1);
  std::cout<<" Thread go boom!"<<std::endl;
  return 0;
}
```

Обратите внимание, что в качестве элемента данных теперь используется экземпляр класса checker, расположенный в динамической памяти. Для ее освобождения была написана функция del_ptr_checker(), указатель на которую помещается в стек очистительно-восстановительных операций. Мы вызываем деструктор объекта при помощи pthread_cleanup_pop(1) – кажется, это должно гарантировать выполнение очистительных действий вне зависимости от способа завершения потока.

Функция del_ptr_checker() сама по себе довольно проста:

```
void del_ptr_checker(void *X){
  checker *del = static_cast<checker*>(X);
  std::cout<<" #-> START del_ptr_checker!"<<std::endl;
  delete del;
  del = 0;
  std::cout<<" #-> END del_ptr_checker!"<<std::endl;
}
```

Мы приводим переданный указатель к типу checker * и вызываем оператор delete для освобождения памяти. del_ptr_checker() определена в файле **helper.hpp**. Функция main() для нашего примера выглядит так:

```
int main(void)
{
  std::cout<<" START TEST #2!"<<std::endl;
  int N=0;
  std::cout<<" Enter N:"; std::cin>>N;
  pthread_t threadA;
  pthread_create(&threadA, NULL, task1, &N);
  pthread_detach(threadA);
  std::cout<<" End MAIN"<<std::endl;
  return 0;
}
```

Приведу результаты выполнения двух тестов в SLAX 6.0.3:

```
START TEST #2!
Enter N:222
Start test thread!
Constructor done! Name:agent
#-> START del_ptr_checker!
Destructor done! Name:agent
#-> END del_ptr_checker!
Thread go boom!
End MAIN

START TEST #2!
Enter N:4589
Start test thread!
Constructor done! Name:agent
#-> START del_ptr_checker!
Destructor done! Name:agent
#-> END del_ptr_checker!
End MAIN
```

Как мы видим, функция del_ptr_checker() вызывается независимо от значения параметра N, задаваемого пользователем, и динамическая память, занимаемая объектом класса checker, всегда корректно освобождается.

А вот что происходит в OpenSUSE 10.1:

```
linux@linux:~/super> ./test_2
 Enter N:654
 End MAIN
linux@linux:~/super> ./test_2
 Enter N:8888
 End MAIN
linux@linux:~/super>
```

Любопытно, но судя по выводу на консоль, поток, использующий динамическую переменную класса checker, даже не создается, не говоря уже о вызове деструктора для объекта. Очень интересный результат! Кстати, аналогичное поведение наблюдается и в Mpentoo Linux 2006.1.

Я тебя породил…

Бывают ситуации, когда одному потоку нужно завершить другой: это может делаться при организации управления программой или для экономии ограниченных ресурсов. В библиотеке *Pthreads* для этих целей предназначена функция pthread_cancel(). В качестве параметра она принимает идентификатор потоковой функции, которую надо завершить, и возвращает 0 в случае успешного выполнения. В уже упоминавшемся документе The Open Group Base Specifications Issue 6 IEEE Std 1003.1, 2004 Edition в разделе, описывающем функцию pthread_cancel(), говорится: *«Функция pthread_cancel() создает запрос на отмену потока. Когда он будет реализован, зависит от текущего состояния потока и его типа. При отмене потока должны быть вызваны обработчики, которые выполнят связанные с отменой подготовительные действия. По завершению последнего обработчика должны быть вызваны деструкторы данных, используемых потоком».*

Посмотрим, что же действительно происходит в данной ситуации, с помощью следующей простой программы:

```
void *task1(void *X){
  std::cout<<" Start thread!"<<std::endl;
  checker Q("spy");
  checker *Z = new checker("agent");
  pthread_cleanup_push(del_ptr_checker, Z);
  pthread_setcancelstate(PTHREAD_CANCEL_ENABLE, NULL);
  std::cout<<" @-> Hello, Threads!"<<std::endl;
  for(int i=0; i<5; ++i)
  {
     Z->say_hello( );
     sleep(1);
  }
  pthread_testcancel();
  pthread_cleanup_pop(1);
  std::cout<<" @-> Logical End of THREAD function"<<std::endl;
}
int main(int argc, char * argv[]){
  cout<<" Start test #3!"<<std::endl;
  int ret;
  pthread_t thread;
  pthread_create(&thread, NULL, task1, NULL);
  ret = pthread_cancel(thread);
  if(ret==0)
  {
     std::cout<<" $$$ Thread CANCEL OK!"<<std::endl;
  }
  pthread_join(thread, NULL);
  std::cout<<" $$$ The thread go boom!"<<std::endl;
  return 0;
}
```

В качестве элемента данных потоковая функция task1() использует два объекта класса checker: один расположен в динамической памяти, второй имеет класс auto. Для уничтожения первого экземпляра мы опять используем стек очистительно-восстановительных операций. Вызов pthread_setcancelstate() разрешает аннулирование нашего потока другим. Функция pthread_testcancel() проверяет наличие необработанных запросов на уничтожение. Если они есть, процесс аннулирования активизируется в точке вызова pthread_testcancel(). Обратите внимание, что в потоковой функции не используются разделяемые переменные и не происходит вызова системных функций (кроме вывода сообщений на консоль), что обеспечивает ее безопасное прекращение.

Результат выполнения тестовой программы в SLAX 6.0.3 таков:

```
Start test #3!
Start thread!
Constructor done! Name:spy
Constructor done! Name:agent
@-> Hello, Threads!
Hello from cheker!
#-> START del_ptr_checker!
Destructor done! Name:agent
#-> END del_ptr_checker!
Destructor done! Name:spy
$$$ Thread CANCEL OK!
$$$ The thread go boom!
```

А вот вывод в системе Mpentoo 2006.1:

```
Start test #3!
$$$ Thread CANCEL OK!
Start thread!
$$$ The thread go boom!
```

Мы видим, что работа потоковой функции, судя по выводу на консоль, завершилась раньше ее начала. При этом объекты класса checker в потоковой функции не создаются.

Давайте проанализируем полученные результаты: для удобства они сведены в таблицу. Ситуации, отмеченные знаком **?**, очевидно, нуждаются в комментариях.

» **Тест № 1**. При запуске программы в OpenSUSE 10.1 наблюдается порядок вызова деструкторов, отличный от всех других систем, получивших +. Однако все деструкторы вызываются, и утечки памяти не происходит.

» **Тест № 2**. В двух системах, отмеченных **?**, наблюдалась неустойчивая работа теста. Она проявлялась в том, что в одном случае результат теста был «правильным», и вывод на консоль полностью соответствовал ожидаемому, а в другом случае наблюдалась проблема с запуском потоковой функции.

Что в итоге?

Какие же выводы можно сделать на основании результатов, отраженных в таблице? Во-первых, при разработке многопоточных

Сводная таблица результатов тестирования

№	Операционная система	Номер теста 1	2	3
1	**Mandriva 2008 KDE LiveCD** Ядро: 2.6.22 GCC: 4.2.2* Glibc: 2.6.1 Libstdc++: 5.0.7 / 6.0.9	+	?	+
2	**Fedora 8 LiveCD** Ядро: 2.6.23 GCC: 4.1.2* Glibc: 2.7 Libstdc++: 6.0.8	+	X	+
3	**ASP Linux 11 LiveCD** Ядро: 2.6.14 GCC: 4.0.2 Glibc: 2.3.5 Libstdc++: 5.0.7 / 6.0.7	+	X	X
4	**Knoppix 5.3.1 LiveDVD** Ядро: 2.6.24 GCC: 4.2.3 Glibc: 2.7 Libstdc++: 5.0.7 / 6.0.10	+	+	+
5	**SLAX 6.0.3 LiveCD** Ядро: 2.6.24 GCC: 4.2.3 Glibc: 2.7 Libstdc++: 6.0.8 / 6.0.9	+	+	+
6	**Ubuntu 8.0.4 LiveDVD** Ядро: 2.6.24 GCC: 4.2.3 Glibc: 2.7 Libstdc++: 6.0.9	+	+	+
7	**Ubuntu 7.10 LiveCD** Ядро: 2.6.22 GCC: 4.1.3 Glibc: 2.6.1 Libstdc++: 6.0.9	+	?	X
8	**ALT Linux 3.0.4 LiveCD** Ядро: 2.6.12 GCC: 3.4.4* Glibc: 2.3.5 Libstdc++: 5.0.7 / 6.0.3	X	X	X
9	**ALT Linux 4.0.3** Ядро: 2.6.18 GCC: 4.1.1 Glibc: 2.5 Libstdc++: 5.0.7 / 6.0.8	+	X	X
10	**MPentoo 2006.1 LiveCD** Ядро: 2.6.16 GCC: 3.3.6 Glibc: 2.3.6 Libstdc++: 5.0.7	X	X	X
11	**Puppy Linux (rus_100) LiveCD** Ядро: 2.6.21 GCC: 4.1.2* Glibc: 2.5 Libstdc++: 5.0.6 / 6.0.8	+	X	X
12	**Gentoo 2008 Beta2 LiveCD** Ядро: 2.6.24 GCC: 4.1.2 Glibc: 2.6.1 Libstdc++: 6.0.8	+	+	+
13	**OpenSUSE 10.1 LiveDVD** Ядро: 2.6.16 GCC: 4.1.0* Glibc: 2.4 Libstdc++: 5.0.7 / 6.0.8	?	X	X

Обозначения:

X – тест выполнен с ошибками, результат не соответствует ожиданиям

? – неожиданный результат выполнения теста, интересен для анализа

+ – тест успешно выполнен, результат адекватен ожиданиям

* – на диске компилятор отсутствует, проверка на бинарной сборке из SLAX

приложений с использованием библиотеки *Pthreads* программист должен уделять повышенное внимание тем фрагментам кода, где используются объекты классов. Несмотря на то, что *Pthreads* имеет средство автоматического освобождения ресурсов – стек очистительно-восстановительных операций, гарантировать обязательность и правильность его использования нельзя. В описании многих функций библиотеки сообщается, что при завершении потока сначала вызываются процедуры из очистительного стека, а потом деструкторы потоковых данных, а на самом деле это не всегда так.

Во-вторых, разработчики Linux-систем вполне осведомлены об особенностях поведения функций библиотеки *Pthreads* при использовании в качестве данных объектов классов. Ведется активная работа в этом направлении, и положительные результаты есть! Показателен пример Ubuntu.

В-третьих, анализ характеристик дистрибутивов, получивших + по всем тестам, обнаруживает, что версия ядра (в этих тестовых образцах) не ниже 2.6.24 (это обязательное условие), библиотека *glibc* – не ниже 2.6.1 (лучше 2.7), библиотека *libstdc++* – не ниже 6.0.8 (лучше 6.0.9), версия компилятора *GCC* – не ниже 4.1.2. Соответственно, дистрибутивы, получившие за тест -, имеют другие версии библиотек. Например, в ALT Linux 4.0.3 используется библиотека *glibc* версии 2.5. Кстати, очень интересно сравнить результаты Gentoo 2008 Beta 2 и Fedora Core 8. Исход их «спора» решила версия ядра. У Gentoo она выше, хотя у Fedora библиотека *glibc* новее. Это говорит о том, что своевременное обновление ядра и ключевых системных библиотек позволяет повысить надежность работы операционной системы.

Читателя наверняка интересует вопрос: а что будет, если программу, скомпилированную в «правильной» системе, например, Knoppix 5.3.1, попробовать запустить в «неправильной», скажем, Knoppix 3.2 RE? Здесь возможны два варианта:

1 Программа не запустится из-за отсутствия необходимых библиотек и выдаст сообщение следующего содержания:

```
knoppix@ttyp0[knoppix]$ ./etalon
./etalon: error while loading shared libraries: libstdc++.so.6:
cannot open shared object file: No such file or directory
knoppix@ttyp0[knoppix]$
```

2 Однако, даже если «правильная» программа запустится в «неправильной» среде, вести себя она будет «неправильно».

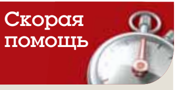
И что делать?

Что можно посоветовать, чтобы свести к минимуму издержки, связанные с особенностями взаимодействия библиотеки *Pthreads* с объектами классов?

Что касается теста №1, то тут может помочь метафора «песочницы»: прием, при котором вся работа с объектами классов, имеющих тип памяти auto (не динамические, а «обычные» переменные), ведется в пределах блока, выделенного в тексте программы фигурными скобками {…}. При выходе из блока происходит автоматический вызов деструкторов, после чего можно «запускать» pthread_exit().

Однако возникает вопрос: что делать, если pthread_exit() вызывается в результате выполнения некоторого условия при работе программы, как, например, в тесте № 2, или происходит аннулирование потока, как в тесте № 3? Здесь относительно универсальным, хотя и достаточно трудозатратным будет такой выход, как «ручное» управление памятью посредством операторов new и delete. Да, это трудно и хлопотно, однако на текущий момент это, наверное, единственный эффективный выход из ситуации. Кстати, для объектов с классом памяти auto деструктор можно вызвать как обычную функцию. LXF

Скорая помощь



Узнать параметры своей системы можно, набрав в консоли следующие команды:

» Версия ядра: uname -a

» Компилятор *GCC*: gcc --version

» Библиотека *glibc*: getconf GNU_LIBC_VERSION

» Библиотека *libstdc++*: ls -l /usr/lib/libstdc++.so.*

Школа LINUX FORMAT



С чем начинать учебный год?



Летний отпуск подходит к концу – самое время вернуться в классы и заняться подготовкой к новому учебному году. **Галина Пожарина** изучает набор открытых программ, которые могут пригодиться в данном процессе.



Наш эксперт

Галина Пожарина
PhD/DCS, директор Центра образовательных инициатив «Открытое программное обеспечение», г. Санкт-Петербург

Лето для образовательных учреждений – не только пора отпусков и ремонта: для инженеров, техников и системных администраторов это время интенсивной работы в компьютерных классах. В течение всего года мы знакомили вас, читатели, с различными свободными и открытыми приложениями – настало время установить их на компьютеры и подготовить к использованию в учебном процессе.

Для реализации образовательных стандартов и учебных программ понадобится следующий набор ПО (**LXF** уже писал о многих из его составляющих; в каждом таком случае в скобках после описания программы приводятся ссылки на конкретные номера и рубрики.)

Для работы с офисными приложениями

» ***OpenOffice.org 2.4.1 Pro***, созданный на базе *OpenOffice.org* российской компанией «Инфра-ресурс» (www.i-rs.ru). Лицензия – GNU LGPL – допускает свободную установку и применение на любом числе компьютеров. Актуальная сборка выполнена для платформ FreeBSD 6.3 и 7.0, GNU/Linux на i386 и x86-64 в пакетах RPM, DEB и Generic, а также Windows в виде самостоятельного дистрибутива, двоичного файла обновлений с версии 2.4.0 и переносимого portable-пакета. Пакет офисных приложений *OpenOffice.org 2.4.1 Pro* содержит все заявленные свойства официальной версии 2.4.1 Community build и дополнен оригинальным набором улучшений от компании «Инфра-Ресурс», ориентированным на русскоговорящих пользователей (**Учебник**, LXF76, LXF80-84, LXF104, LXF105).

Для изучения графики

» ***GIMP*** (*GNU Image Manipulation Program*) – растровый графический редактор, т.е. программа для создания и обработки растровой графики с частичной поддержкой векторных операций. Типичные задачи, которые можно решать при помощи *GIMP*, включают создание графики и логотипов, масштабирование и кадрирование фотографий, раскраску, комбинирование изображений с использованием слоев, ретуширование и преобразование графических файлов в различные форматы (**Учебник**, LXF70-73, LXF106 и далее).

» ***Blender*** – пакет для трехмерного моделирования, включающий средства анимации, рендеринга, постобработки видео, а также создания интерактивных игр. Особенностями *Blender* являются малый размер, высокая скорость отрисовки, наличие версий для множества операционных систем: FreeBSD, GNU/Linux, Mac OS X, SGI Irix 6.5, Sun Solaris 2.8 (sparc), Microsoft Windows, SkyOS, MorphOS и Pocket PC. Пакет предоставляет такие возможности, как динамика твердых тел, жидкостей и «мягких тел», система горячих клавиш, большое количество легкодоступных расширений, написанных на языке Python (**Учебник**, LXF87/88-LXF91; **Игрострой**, LXF100/101, 103, 105, 107).

» ***Scribus*** – настольная издательская система, созданная для пользователей Linux/Unix/Mac OS X и Windows и аналогичная по концепции *Adobe InDesign* или *QuarkXPress*. Пакет может использоваться для создания макетов бюллетеней, корпоративных циркуляров, постеров, учебных материалов, технической документации, визиток и других документов, требующих гибкой визуальной верстки и серьезных возможностей по обработке изображений, а также точного управления типографикой и размерами иллюстраций, каковых нет в обычных текстовых процессорах; при создании документов для высококачественной тиражируемой печати, файлов, распространяемых через Интернет в формате PDF и презентаций, интерактивных PDF-документов с заполняемыми формами (LXF85, LXF96).

» ***Inkscape*** – векторный графический редактор, удобен для создания как художественных, так и технических иллюстраций. Он будет полезен при разработке презентаций, логотипов и плакатов, а также схем, графиков, баннеров и макетов сайтов, пиктограмм для приложений и кнопок, графики для игр (**Учебник**, LXF74/75-79, LXF81-86).

Для работы с векторной графикой возможно также использование *OpenOffice.org Draw*, который входит в офисный пакет *OpenOffice.org* (см. выше).

Для обучения алгоритмизации и программированию

» ***Code::Blocks*** – кроссплатформенная среда разработки, которая поддерживает следующие компиляторы: GCC (MinGW для Windows), Microsoft Visual C++ Compiler, Digital Mars, Borland C++ (версии 5.5), Open Watcom и Sun DCC. Среда сама передает компилятору все ключи, необходимые для его работы с заданными опциями. Компиляция может осуществляться как напрямую, так и через make-файлы. В *Code::Blocks* имеется удобный и многофункциональный редактор кода, поддерживающий подсветку синтаксиса и фолдинг (сворачивание) блоков. Среда работает как с исходными текстами на C/C++, так и с XML-файлами. Имеется проводник по входящим в проект классам, а также список всех используемых переменных, констант, классов и пространств имен, дерево ресурсов и список просматриваемых во время отладки переменных (**Школа**, LXF105).

» ***Скретч* [Scratch]** – среда программирования для начинающих. Позволяет создавать собственные анимированные и интерактивные истории, игры и другие произведения, которыми можно обмениваться внутри международной среды, постепенно формирующейся в сети Интернет. *Скретч* базируется на традициях языка Лого и написан на языке Сквик. В *Скретче* можно создавать фильмы, играть с различными объектами, видоизменять их, перемещать по экрану, устанавливать формы взаимодействия между объектами. *Скретч* – объектно-ориентированная среда. Начальный уровень программирования столь прост и доступен, что она рассматривается в качестве средства обучения не только старших, но и младших школьников (Школа, LXF105).

» ***Free Pascal*** – свободно распространяемый компилятор языка Паскаль. Free Pascal поддерживает различные аппаратные архитектуры (Intel x86, Amd64/x86_64, PowerPC, Sparc) и операционные системы (Linux, FreeBSD, Mac OS X/Darwin, Mac OS classic, DOS, Win32, OS/2, Netware и MorphOS). Синтаксис *Free Pascal* полностью совместим с Turbo Pascal 7.0, а также, по утверждению разработчиков, с большинством версий Object Pascal/Delphi (классы, информация о типах времени выполнения – RTTI, исключения, ANSI-строки, Wide-строки, интерфейсы). Для пользователей Apple есть режим совместимости с Mac Pascal. *Free Pascal* поддерживает перегрузку функций, операторов, глобальные свойства и некоторые другие особенности. Компилятор лицензирован по GNU GPL, а идущие вместе с ним пакеты и библиотеки – по модифицированной LGPL, что позволяет распространять собранные вами программы на любых удобных для вас условиях.

Для обучения работе с глобальными ресурсами Интернета

» ***Mozilla Firefox 3***, вышедший 17 июня 2008 г. В новой версии браузера обновился движок – улучшилось качество отображения сайтов, возросла скорость их загрузки, повысилась совместимость со стандартами. *Firefox 3* содержит массу крупных и мелких изменений интерфейса, и, как следствие, работать стало намного комфортнее.

» ***Opera 9*** – «самый быстрый браузер на Земле». Самой сильной стороной *Opera* является работа со сценариями JavaScript – по некоторым данным, это происходит примерно вдвое быстрее, чем в других браузерах. В *Opera* встроен TDI-интерфейс (вкладки), настраивается блокировка всплывающих окон, есть защита от мошенничества, менеджер закачек, BitTorrent-клиент, меню поиска, RSS-агрегатор. Одна из особенностей браузера – возможность быстрого перехода к наиболее часто посещаемым страницам, что значительно облегчает навигацию по web-сайтам.

» ***Mozilla Thunderbird 2*** – программа для работы с электронной почтой и новостными группами. Поддерживает протоколы SMTP, POP3, IMAP, NNTP, RSS-ленты. *Thunderbird* работает в Microsoft Windows, Mac OS X и Linux, причем набор возможностей одинаков на всех платформах.

Для младших школьников

» ***GCompris*** Данный образовательный пакет состоит из пятидесяти различных игр, сгруппированных по восьми категориям. Приложения адресованы прежде всего детям дошкольного и младшего школьного возраста; персонажи яркие и привлекательные, имеется звуковое сопровождение. Пакет доступен как для Linux, так и для Mac OS X и Windows (**Школа**, LXF107).

» ***Tux Paint*** – среда для рисования, адресованная в первую очередь младшим школьникам. Интерфейс программы больше напоминает настольную игру. Организация панелей инструментов эргономична и интуитивно понятна. Изначально *Tux Paint* был создан для Linux, но сейчас доступен для Microsoft Windows, BSD, OS X и Solaris, а также BeOS. Поддерживается русский язык. Яркий и привлекательный редактор в доступной форме представляет основные функции компьютерной обработки изображений (**Школа**, LXF106).

В помощь при изучении математических дисциплин

» ***Maxima*** – система символьных и численных вычислений, поддерживающая дифференцирование, интегрирование, разложение в ряд, преобразование Лапласа, решающая обыкновенные дифференциальные уравнения и системы линейных уравнений, работающая с многочленами, множествами, списками, векторами, матрицами и тензорами. *Maxima* производит численные расчеты высокой точности, используя точные дроби, целые и числа с плавающей запятой. Система позволяет строить графики функций и статистических данных в двух и трех измерениях. Исходный код *Maxima* может компилироваться на многих системах, включая Windows, Linux и MacOS X (**Учебник**, LXF81-86).

» ***Scilab*** – пакет научных программ для численных вычислений, предоставляющий мощное открытое окружение для инженерных и научных расчетов. *Scilab* содержит сотни математических функций с возможностью добавления новых, разнообразные структуры данных (списки, полиномы, рациональные функции, линейные системы), интерпретатор языка высокого уровня. В системе доступны такие инструменты, как 2D и 3D-графики, анимация, разреженные матрицы, полиномиальные и рациональные функции, дифференциальные и прочие оптимизации. *Scilab* имеет схожий с *Matlab* язык программирования; в его составе имеется утилита, позволяющая конвертировать документы *Matlab* в *Scilab* (**Учебник**, LXF106-109).

Если вы уже работаете в Linux (а мы надеемся, это так и есть), рекомендуем широко использовать образовательный пакет KDE Edutainment Project (*KDEEdu*), который является частью рабочего стола KDE. В *KDEEdu* входят программы для преподавателей, помогающие планировать уроки, а также достаточно большой набор приложений для детей в возрасте от 3 до 18 лет.

Для организации системы дистанционной поддержки учебного процесса

При наличии подключения к Интернету в образовательных учреждениях могут использоваться соответствующие системы управления контентом (Content Management System, CMS).

» ***Atutor*** Бесплатная и открытая основанная на Web система управления контентом обучения (LCMS). Администратор способен инсталлировать или обновить *ATutor* за минуты. Преподаватели могут быстро собрать, упаковать и перенести учебные курсы, а также открыть их в сети. Студенты получают адаптируемую среду обучения. В настоящее время доступна версия 1.6.1, поддерживающая русский язык (**Школа**, LXF104).

» ***Moodle 1.9.2*** Modular Object-Oriented Dynamic Learning Environmemnt [модульная объектно-ориентированная динамическая учебная среда] реализует философию «педагогики социального конструкционизма» и ориентирована, прежде всего, на организацию взаимодействия между преподавателем и учениками, хотя годится и для поддержки традиционных дистанционных курсов, а также сопровождения очного обучения.

Таким образом, использование кроссплатформенного открытого программного обеспечения не только сделает работу в новом учебном году эффективной и интересной, но и подготовит преподавателей и учащихся к полному переходу на Open Source, что уже сделали наши коллеги во многих странах Европы. LXF



мой первый язык

Многие из нас начинали изучать программирование с Pascal или Basic – но действительно ли это оптимальный выбор? **Светлана Шапошникова** представляет нового претендента на роль первого языка программирования.



Наш эксперт

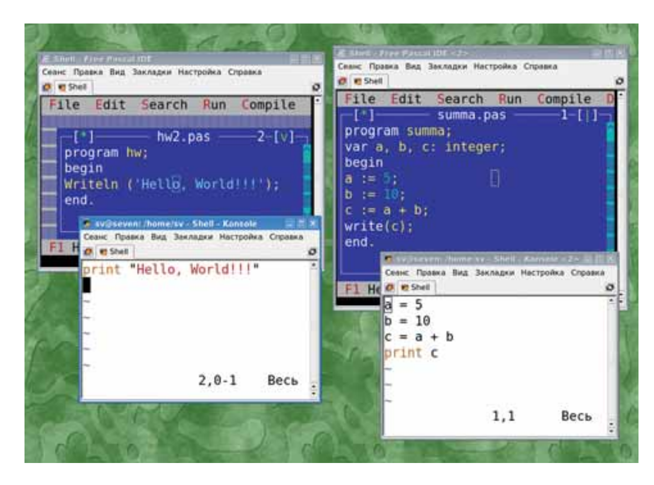
Светлана Шапошникова
педагог дополнительного образования кружков компьютерной техники ЦД(Ю)ТТ «Городской» г. Липецка, использует Linux в образовательном процессе с 2005 года.

Большинство российских школьников и студентов начинают изучать основы программирования, используя язык Pascal. Возможно, когда-то это был действительно разумный выбор, но в настоящее время он, скорее, дань привычке и комплексу старой методической литературы. Сегодня, когда учебным заведениям предстоит миграция на Linux, разработчики большинства «школьных» дистрибутивов включают в состав своих продуктов компилятор *Free Pascal* (www.freepascal.org) и среду Lazarus (www.lazarus.freepascal.org). Понятно, что это скорее необходимость, делающая переход на новую систему более спокойным. Но насколько она уместна, если исходить из позиции, что методические разработки – дело наживное? Нет ли языка, более родного для Linux и в то же время не уступающего Pascal'ю по своим образовательным возможностям (а ведь Pascal был создан для обучения программированию)?

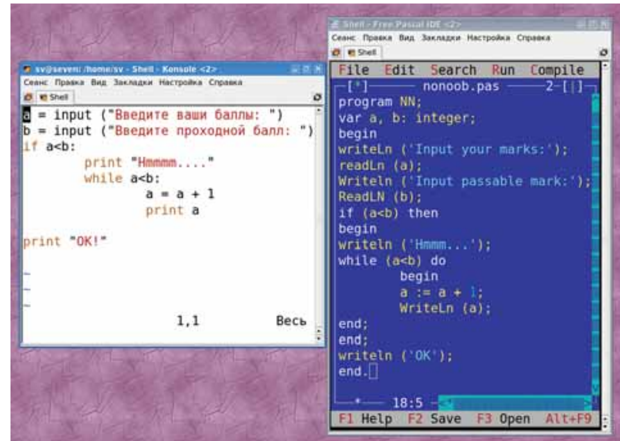
Если вам доводилось работать с детьми и обучать их основам программирования (да, в принципе, неважно чему), то вы могли заметить следующее: детская психология при изучении нового материала весьма требовательна к наглядности, концентрации на главном и сведению к минимуму иных «раздражающих» факторов. Кроме того, сегодняшние дети начинают изучать программирование несколько раньше, чем десять лет назад. Зачастую предмет «Информатика» уже присутствует в третьем-четвертом классах, а следовательно, к программированию можно подойти уже к пятому-шестому. 11-12-летние дети могут быть психологически не готовы воспринимать то, что вполне приемлемо для 15-летних. Например, обилие числовых типов данных в Pascal, «громоздкий» синтаксис языка. Хочется, чтобы первый язык программирования был достаточно прост и ясен. Конечно, для средней школы может подойти Лого, однако он более ориентирован на младшеклассников, а также чаще используется при изучении темы «Алгоритмика», а не «Программирование». Лого – хороший заменитель алгоритмического языка А.П. Ершова (широко используемого на уроках информатики в 80-90-х годах) для детей младшего школьного возраста. Если же говорить о первом настоящем языке программирования, то, возможно, следует обратить внимание на язык Python (python.org). В русском языке это имя часто читают как «Питон», но вообще-то правильное произношение – английское, «Пайтон».

Почему Python?

Python был создан Гвидо ван Россумом [Guido van Rossum] примерно в 1991 г. Его отличительной особенностью является простой и ясный синтаксис. С одной стороны, вспомогательные элементы языка сведены к минимуму: это позволяет делать меньше ошибок на начальном этапе обучения программированию и быстрее писать программы. С другой стороны, Python жестко регламентирует «правила отступов», поэтому воспитание аккуратности и внимательности не пострадает. Учащиеся привыкнут к хорошему стилю программирования. Чтобы преимущества Python стали очевидными, достаточно сравнить код небольшой программы, написанной на нем и на Pascal (**рис. 1**). Большое количество



› Рис. 1. Pascal и Python: почувствуйте разницу.



› Рис. 2. Строгий синтаксис Python заставляет ученика правильно оформлять код.

«begin-end» в сочетании с необязательностью соблюдения отступов делают проверку программы, написанной на Pascal, мукой. Кроме того, приучив ученика к строгости стиля Python, мы вынуждаем его видеть и понимать управляющие структуры (**рис. 2**).

Если при написании программы с использованием языка Pascal каждая переменная должна быть обязательно объявлена перед своим использованием, то в Python ей можно просто присвоить значение, которое и определит ее тип. Хорошо это или плохо – вопрос неоднозначный. Предположим, что для профессионального программиста и даже для старшеклассника это недостаток. В первом случае может проявиться непредвиденный результат или ошибка, а во втором – недостаточное понимание обработки компьютером различных типов данных. Однако при изучении начал программирования, отсутствие необходимости отслеживать многообразие числовых типов позволит уделить больше внимания логике программирования и управляющим структурам.

В пользу Python как первого языка программирования говорит и то, что он является интерпретируемым. На начальных этапах изучения языка рассматриваются отдельные небольшие конструкции, осваивается синтаксис; а значит, не обязательно создавать файл и сохранять код – это только отвлекает. В режиме интерпретатора можно, набрав команду, тут же получить ответ. Ученик сразу видит результат работы или допущенную ошибку, а следовательно, быстрее и легче приходит к пониманию происходящего.

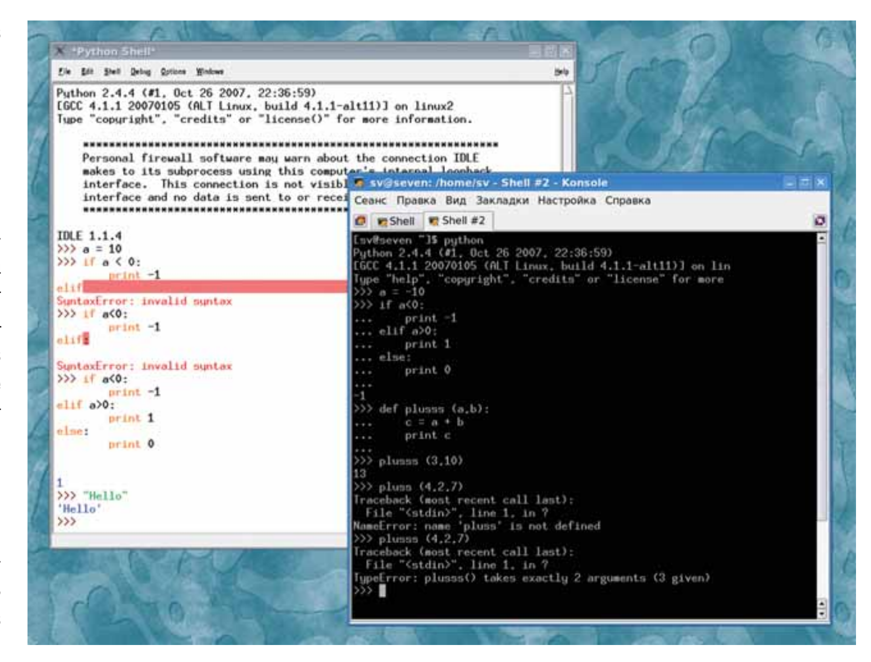
А где удобства?

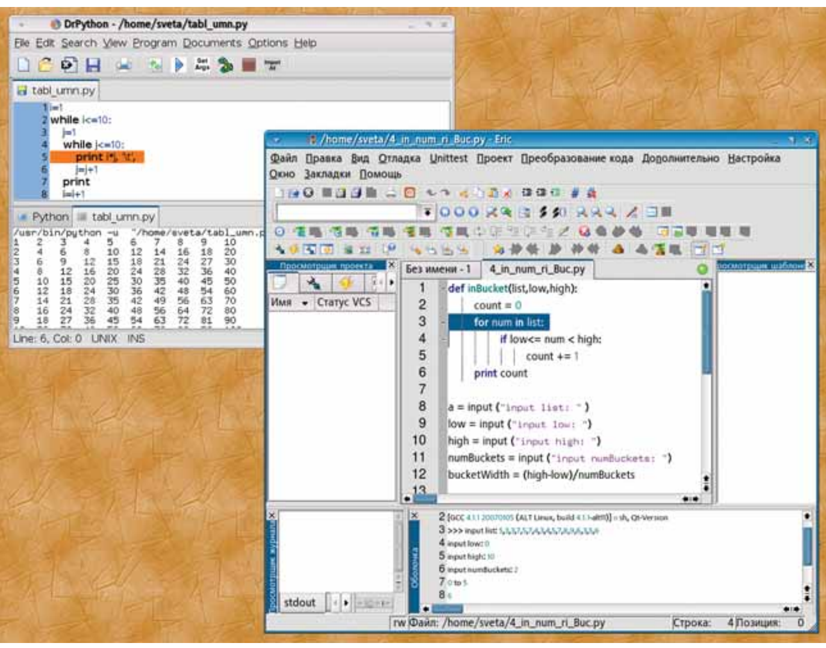
В качестве одного из достоинств Pascal некоторые авторы отмечают наличие у него интегрированной среды программирования, включающей редактор, компилятор, компоновщик, отладчик и др. С Python не все так однозначно. Чтобы начать работать в интерактивном режиме, следует набрать команду **python** в консоли. Кроме этого, существует специальная интегрированная среда разработки IDLE (в дистрибутиве для Windows она присутствует по умолчанию, в Linux требуется доустановить пакет). Преимущество консоли – в возможности прокручивать историю команд, а IDLE – в подсветке синтаксиса. Но поскольку в режиме интерпретатора более-менее длинных программ все равно не напишешь, то первое преимущество, думается, значительнее. Для создания же скриптов [так часто называют программы на Python, второй вариант – «сценарии». Скрипты Python обычно имеют расширение .py, хотя это совершенно не обязательно, – *прим. ред.*] пригодно большинство текстовых редакторов Linux (не считая IDLE, в которой тоже можно набирать и сохранять файлы). Чтобы выполнить скрипт, достаточно дать команду **python путь/имя_программы**. И, наконец, для Python существуют специальные (содержащие большее количество инструментов, чем IDLE) среды разработки: например, *DrPython* (**drpython.sourceforge.net**) и *Eric* (**www.die-offenbachs.de/eric**) – см. **рис. 3**. И если первая ориентирована на школьников, то вторая больше подходит для серьезных задач. Помимо прочего, синтаксис Python поддерживает и *KDevelop*, так что каждый может выбрать «на свой вкус и цвет».

А дальше?

Наконец, нельзя не сказать о том, что Python – это полноценный универсальный язык программирования. Освоите азы – всегда будет, куда двигаться дальше. Python поддерживает функциональное и объектно-ориентированное программирование, для него существует большое количество библиотек, в том числе – для создания графического пользовательского интерфейса.

Каждый язык программирования имеет свои особенности и области наиболее частого применения. В качестве первого можно выбрать почти любой, однако не каждый из них будет эффективным. Также при выборе следует учитывать возраст обучаемых и род задач. Так, если цель – познакомить учащихся среднего школьного возраста с современным и используемым настоящими (вроде как бывают и игрушечные?) программистами языком, то на эту роль может прекрасно подойти Python.





› Рис. 3. Среды разработки для Python: выбирай на вкус.

Содержание образования меняется, должны меняться и его «инструменты». Следует, конечно, учитывать и человеческий фактор: многим учителям с большим стажем и энным количеством методических разработок по Pascal или Basic трудно перейти на что-то новое (да и приемные комиссии некоторых ВУЗов еще требуют от абитуриента знания именно Pascal). Поэтому включение в «школьные» дистрибутивы компилятора FreePascal сегодня является необходимостью. Однако начинающие учителя информатики, не обремененные опытом, возможно, найдут Python весьма подходящим языком для обучения основам программирования. LXF

Узнайте больше!



Заинтересовались? **Linux Format** публиковал уроки Python в выпусках LXF74/75–84. Вы можете загрузить их в электронном виде с сайта журнала: **www.linuxformat.ru/pdf.phtml**.



» Записал
Евгений Балдин



» **Кто**
Александр Сергеевич Добдин, математик-программист, заведующий лабораторией микро-ЭВМ.
» **Откуда**
Саратовский государственный университет им. Н.Г.Чернышевского, механико-математический факультет.
» **Проект**
Перевод компьютерной сети факультета на открытое ПО.
» **Результат**
Все работает, в том числе и потому, что процесс перехода является эволюционным.

Математики из Саратова выбирают GNU/Linux

В точных науках вопрос: «Что использовать в качестве прокладки между программным и аппаратным обеспечением?» уже давно не является насущным: конечно, Linux. Возможно, не прямо сейчас, но когда руки дойдут – как это было в случае **Александра Добдина**.

В Саратовском государственном университете (СГУ) я работаю с 2001 года. Начинал дежурным в компьютерных классах механико-математического факультета еще будучи студентом второго курса. С 2008 года возглавляю лабораторию микро-ЭВМ СГУ. Мы занимаемся ИТ-сопровождением учебного процесса на факультете.

О Linux я услышал из рассказов своего друга Сергея Кудинова еще в 1999 году, но увидеть эту систему в реальности удалось только в 2001. Прежде чем поставить его, я проштудировал «Путь к LINUX» В. Водолазского. Первым дистрибутивом был Mandrake (вроде бы 7.2), выбранный по принципу «что есть, то и ставим». Впечатления (после Windows) оказалось положительными. KDE 2 тех лет была уже красивее Проводника, хотя пришлось немного попривыкать к совсем другим «рюшечкам». Серьезно же заинтересовался Linux в процессе обучения в университете, проработав год в лаборатории микро-ЭВМ (2002 г.).

Как настольная ОС, Linux появился в СГУ в феврале 2004 года в одном из компьютерных классов мехмата. К тому времени он уже применялся на некоторых серверах. Занятия по нему (две пары в неделю) поначалу велись по специальному курсу доцентом кафедры математической физики и вычислительной математики О.Б. Горбуновым

К началу 2006/2007 учебного года на базе опробованных ранее дистрибутивов был подготовлен большой локальный сборник пакетов, включающий в себя множество аналогов несвободного ПО. Удалось заменить все программы, кроме тех, что были жестко прописаны в государственном образовательном стандарте для ряда специальностей. Поскольку на тот момент в городе уже начались проверки различными ведомствами, то факультетом было принято решение отказаться от использования нелицензионного ПО и заменить его аналогичными бесплатными и, по возможности, свободными продуктами.

Что касается конкретно Linux, то на нашем факультете из 12 кафедр целиком и полностью на него перешла только одна: кафедра компьютерной алгебры и теории чисел. Linux установлен и на двух из трех факультетских серверов, а в масштабах университета открытая ОС обслуживает почту и файлы, сервисы DNS и DHCP. Дистрибутивы самые разные, в основном, Fedora, OpenSUSE и Gentoo; как такового, стандарта нет.

Сегодня все ПО, используемое факультетом, легальное. Доля свободных программ в учебном процессе составляет около 90%. Нам также пришлось закупить лицензии и на некоторые коммерческие продукты, например, графические пакеты для специальности «Web-дизайн»: причины были изложены выше. До сих пор

не удалось найти аналогов для некоторых пакетов имитационного моделирования – обходимся демо-версией I Think. Преподаватели пока еще спрашивают о совместимости *Octave* и *Scilab* с *Matlab*. Дело движется в нужном направлении, хотя поначалу большинство преподавателей не оценило «шутку юмора». К счастью, многие уже тогда понимали, что будущее за открытым ПО, поэтому поддержали наши начинания. С нового 2008/2009 учебного года будет организован семинар: сначала для преподавателей и сотрудников, а затем и в более широком формате, для всех желающих.

В компьютерных классах мехмата (около 50 машин, от Celeron 800 до Pentium D) установлен Gentoo Linux. Причины – «исторические»: первый опыт был получен именно на нем. Тогда нам очень хотелось разобраться в системе досконально – это сейчас я понимаю, что чем лучше узнаешь Linux, тем больше понимаешь, как мало знаешь. Компьютеры у нас старые, поэтому (чтобы выжать из них максимум) все пакеты собираются под конкретный процессор. В силу специфики образовательной деятельности для поддержки компьютерных классов было решено создать собственный локальный репозитарий, благо в Gentoo это несложно.

После некоторых обсуждений и проб мы остановились на следующей схеме: есть гостевая учетная запись для студентов (student), настройки которой позволяют сразу же приступить к работе. Домашняя директория, если ее не существует, создается при каждом входе в систему (модуль *pam_mkhomedir*). Затем студент переписывает свои рабочие файлы и настройки программ с университетского FTP-сервера с использованием личной учетной записи. После завершения последнего сеанса в консоли домашний каталог пользователя student удаляется. Это удобно: на рабочих станциях не скапливаются временные файлы, и не нужно следить за свободным пространством на разделе **/home**. При этом на случай аварийного завершения работы компьютера (бывает, что «балуются» электричеством) данные пользователя на диске не будут утеряны. Для обмена файлами предусмотрен временный каталог на сервере, доступный по NFS: он очищается ежедневно. В ближайшем будущем планируем централизованную аутентификацию плюс монтирование домашнего каталога через NFS. Таким образом, пользователю не придется копировать свои настройки и рабочие файлы с FTP, и он сможет прозрачно работать за любым компьютером.

Согласно регламенту, раз в полгода (перед началом очередного семестра) в компьютерных классах производится обновление образа всей системы до некоторого промежуточного состояния. Это происходит в несколько этапов:

1. На «эталонной» машине собирается новая или обновляется предыдущая версия системы с оптимизацией под самые слабые ПК;
2. С помощью дистрибутива, загружаемого по сети и небольшого скрипта монтируются локальные и удаленные файловые системы. Далее создается tar-архив иерархии каталогов сервера. Его объем составляет порядка 4,5 ГБ;
3. После создания архива все клиентские компьютеры загружаются через PXE в ту же вспомогательную систему. Посредством SSH на каждой станции запускается скрипт, который

- в зависимости от выбора, создает/изменяет разделы на жестком диске и форматирует их;
- распаковывает архив эталонной файловой системы;
- берет из базы данных (пока это текстовый файл, но позже планируется перейти на использование полноценной СУБД, например, *MySQL*) имя компьютера и копирует конфигурационные файлы для данного типа машин (согласно MAC-адресу);
- записывает загрузчик в MBR и перегружает компьютер.

Поскольку пропускная способность нашей локальной сети составляет всего 100 Мбит/сек, таким образом лучше обновлять не более пяти машин сразу. Для компьютерного класса это выливается в от 1 до 3 часов работы. Для администрирования клиентских станций используются SSH и *sudo*, настроенные для одного специального пользователя без ввода пароля. Например, если необходимо выполнить установку нового ПО в классе, создается небольшой скрипт (см. **LXFDVD**); он и выполняет всю работу.



Такова техническая составляющая. Основные же проблемы – это, конечно, страх перед чем-то новым и «вдруг не получится». Это касается и преподавателей, и студентов. Плюс нежелание читать документацию, хотя ее в нашем случае и искать не надо, т.к. страница с ней (oss.mexmat.sgu.ru) запускается автоматически вместе с графической оболочкой. В целом, перевод учебного процесса на Linux можно считать завершенным и удачным. Остается только несколько курсов, за которые отвечают преподаватели, на дух не переносящие среды разработки для Linux. Для таких «особенных случаев» пришлось закупать подписку на Microsoft Academic Alliance, в которую включена Visual Studio.

В дальнейшем мы планируем расширить набор приложений для научной работы и обучения. Хотелось бы создать на базе Университета центр разработки открытого ПО с привлечением студентов: у нас есть контакты с компанией ALT Linux. Сейчас в нескольких регионах идет эксперимент по внедрению Linux в школах по национальному проекту «Образование». Наш университет на базе Представительств СГУ по области и уже имеющегося опыта собирается установить открытую ОС в 13 школьных компьютерных классах. На семинарах и курсах планируется обучать талантливых учеников 10–11 классов школ районов Саратовской и Волгоградской областей, которые бы могли поддерживать данное ПО на местах. Мы следим за развитием системы, своевременно обновляем зеркала.

На этом пути нам не хватает времени, хорошо обученных сотрудников, ну и финансирования (ставок на дежурных и администраторов в компьютерных классах), естественно. Всю факультетскую сеть (порядка 120 машин) обслуживают два администратора, т.е. построенная нами система держится на энтузиазме сотрудников. Кстати, так было всегда. За примером далеко ходить не надо – это Вычислительный центр СГУ (http://www.cc.sgu.ru) основанный на том же энтузиазме и интересе людей в далеком 1957 году. Столько лет прошло, а нам до сих пор интересно. LXF

Благодарность

Пользуясь случаем, хочу отметить, что все вышеизложенное было бы невозможно без замечательных сотрудников нашего факультета и, в частности, без помощи со стороны доцентов Вячеслава Петровича Склярова, Александра Юрьевича Трынина и Юрия Анатольевича Блинкова.



» Записал Евгений Балдин



» **Кто**
Андрей Николаевич Филинов, дипломированный инженер-механик, учитель информатики высшей квалификационной категории.
» **Откуда**
Муниципальное образовательное учреждение «Гимназия» им. С.В. Ковалевской, г. Великие Луки, Псковская область.
» **Проект**
Жизнь преподавателя информатики.
» **Результат**
«Так я стал апологетом GNU/Linux».



Когда мне все это надоело...

Жизнь учителя непроста. Как может быть легко тому, кто из потенциального человека разумного делает человека социального? Зато ее можно сделать интересной – и **Андрей Филинов** точно знает, как именно.

Давным-давно, еще в прошлом тысячелетии, в каком-то журнале мне полюбилась рубрика «Когда мне все это надоело…», в которой автор с неподражаемым юмором описывал свои злоключения и решения компьютерных проблем. Главное было не то, что эти проблемы решались, а то, что автор дарил тебе крылья и говорил: «Лети!». В смысле твори сам, тебе все по плечу, ты сможешь, как я, и даже лучше…

За 18 лет преподавания информатики многое надоело и мне, причем больше всего раздражали не глобальные проблемы, а мелочи. Ну, вот, например, вы установили чистенькую, еще пахнущую свежей краской Windows на все школьные ПК, любовно настроили ее, привели в порядок рабочий стол, налепили ярлычков для детишек, чтоб те не мучились и не искали «Главную Кнопку» – и что? Буквально через неделю все машины кишат вирусами, диски под завязку забиты какими-то песенками, картинками… А где ярлыки? А для чего тебе этот дурацкий сверхнаклонный шрифт? Что? Тебе так больше нравится? Ну, допустим, на вкус и цвет, как говорится… А где Delphi? Как: «Только нажал вон ту кнопочку»?!

И пошло, и поехало… Они ломают, я чиню, они ломают… Стоп! Решаем проблему радикально: создаем аварийный CD для восстановления – пять минут, и Windows сияет первозданной чистотой. А кто это нашалил на четвертой машине, кто написал на рабочем столе плохое слово? Вот заведу я персональные учетные записи, и тогда посмотрим! А смотреть оказалось нечего. Общедоступные системные папки все так же плодятся в геометрической прогрессии, как кролики, программы хотят компилироваться только с правами администратора, да и сама система живет только по ей одной известным законам…

Но каплей, переполнившей терпение, были три письма от нашего образовательного начальства с требованием незамедлительно стереть пиратское ПО (на дворе март 2007 г., дело Поносова потрясло «пиратскую Россию»).

Что? Меня – еще и в тюрьму? Отберут старенькие «Жигули» с розовой мечтой о тихом частном домике, подаренным бабушкой?

Так я стал апологетом Linux.

За год работы с ним в школе я вернулся в старые добрые времена DOS. Не в смысле примитивности, а в смысле осознанности всех действий. Скопировал файл – он тут же появился в каталоге. Удалил – он действительно исчез совсем, будь он хоть трижды системным: Linux лишь тихонечко вздохнет перед расставанием. Вот главное чувство: ты полный хозяин своего электронного помощника. Надо – он и на верную смерть пойдет по твоему приказу!

Ах, как он вырос за последние пять лет, этот Linux! Трехмерный рабочий стол, невероятное языковое пиршество, великое множество удобных и мощных программ, практически уже существующая совместимость с Windows благодаря *Wine*...

Я окунулся в мир справедливости и дружелюбия: в стране Linux вам всегда рады, вас любят, о вас помнят и заботятся. Да! Такое бывает только в семье. Вернулись крылья, казалось, уже навсегда потерянные.

> «Лучше день потерять, зато потом за пять минут долететь.»
> *«Крылья, Ноги, Хвосты.»*

Кстати, о крыльях – которые ускоряют. Я перепробовал множество различных дистрибутивов: Mandriva, Fedora, Ubuntu и еще десяток



других, и понял, что справиться самостоятельно с требованиями к учебному процессу трудно. Но тут я заметил, что в стране пингвинов случаются чудеса: совершенно безвозмездно прислали аж из Франции два Ubuntu, остальное можно качать с сайтов до одурения. Вы можете себе представить, что Билл Гейтс разговаривает с вами где-нибудь во Пскове, дарит свою Vista, отвечает на письма?.. Сказка! Не пойму только, страшная или не очень. А вот встретиться с генеральным директором ALT Linux А.В. Смирновым мне довелось, и почти у себя дома. Каюсь, выпросил автограф (не для себя, для школьников) и получил в подарок множество всяческих Linux'ов (9 штук!), среди которых был *один*, да какой! И сред программирования в нем великое множество, и офис, и запись дисков, *Wine*, и… Но главное, что умел ALT Linux – это загружать по сети удаленные компьютеры и превращать их в терминалы. Все в этом мире развивается по спирали, хотя некоторые говорят, что по кругу. В 1986 году я, чертыхаясь, тыкал кнопочки на терминале ЕС ЭВМ и никак не предполагал, что через 20 лет буду мечтать о подобной системе.

Школа стала лавинообразно насыщаться техникой. 50 компьютеров – это 50 операционных систем и 500 прикладных программ, которые постоянно зависают, ломаются, устаревают или доустанавливаются в 50-ти экземплярах. Нет, конечно спасибо Д.А. Медведеву и национальному проекту, но *такое* количество техники, как за один только 2007/2008 учебный год, мы не получали никогда. Не то чтобы мы не были готовы переварить этот огромный кусище из принтеров-сканеров, интерактивных досок и компьютеров. Дело скорее в маленьком школьном штате, где учитель – это и потребитель, и лаборант, и инженер, и web-дизайнер, и типографский работник, и администратор, и зам. по ИКТ в одном лице. Даже если будут платить все эти ставки (крохотные такие, на выживание дюймовочек рассчитанные), а их еще и раздергивают, как дополнительную оплату для других малюсеньких ставочек в бухгалтерии, секретарю… И в сутках, между прочим, только 24 часа: все не успеть. А кто школьный сайт наполнять будет, а кто буклеты и дипломы печатать? Тут уж как повезет школьному «Главному-По-Компьютерам»: на какой ступени освоения техники находятся главные и не очень начальники, какие задачи они перед тобой ставят в силу их, этих задач, понимания. А то можно и тексты набирать всем и вся, откликаясь на просьбы-требования: «А кто мне почту посмотрит?» или «Распечатайте мне погоду на неделю».

Уффф, дошел до крыльев, которые ускоряют. Устанавливаем ALT Linux Terminal, настраиваем сетевые карты на загрузку по сети и наслаждаемся тишиной, в смысле временным затишьем. 15–20 машин работают, а ухаживать надо только за одной, за сервером. Чувствуете разницу? Не 50 компьютеров обслуживать, а всего один-два! Конечно, это только после запуска терминального сервера можно так просто об этом говорить. Сам же процесс нетривиален, как и все в нашей странной школьной работе, где мы сеем разумное, доброе, вечное. Подробнее? Хорошо. Еще подробнее, пошагово, с картиночками? Да нет проблем! Пишите – всегда отвечу [адрес в редакции, – *прим. ред.*].

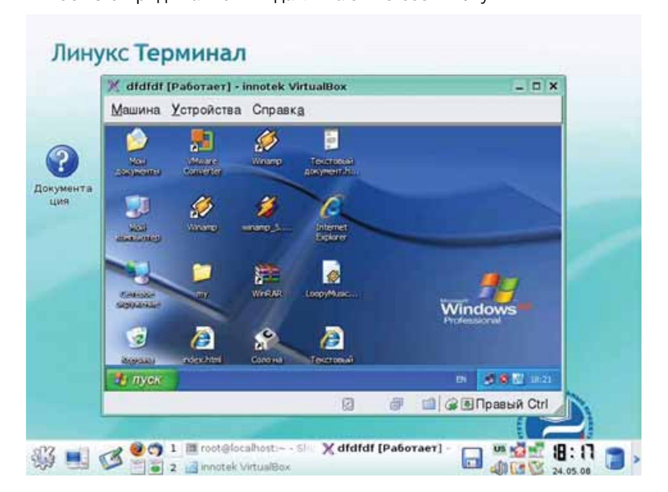
Хорошая новость в том, что ALT Linux Personal Desktop 4.0 + LTSP 5 устанавливается на практически любое оборудование. Но это все же сервер! Не забывайте, что чудес не бывает, и для данных целей необходимо выделить *самую мощную машину* в школе. Это значит – не давать никому на ней работать, как на ПК, даже себе запретите. Установили сервер, запустили – и не выключайте его вообще никогда. И не пихайте в него новые программы без острейшей на то необходимости и только после обкатки на другом, испытательном сервере. Что, нет машин? А 15 клиентов, которым теперь не нужны их железные внутренности?

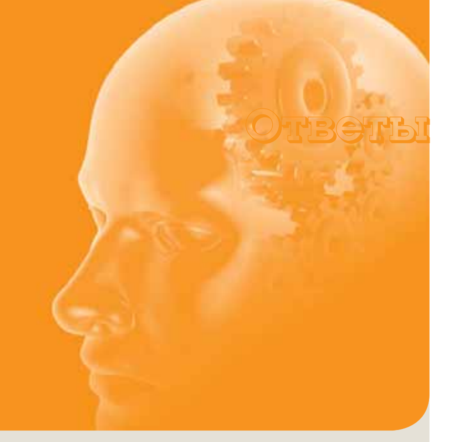
Вот все ушли по домам, тихо и спокойно в родной школе, никто и ничто не отвлекает, а вы на любой клиентской машине восстанавливаете загрузку с винчестера, устанавливаете копию терминального сервера и можете мучить его, испытывать, экспериментировать, только не трогайте настоящий – так он дольше проработает. Про годы не скажу, а семь месяцев Fedora 8 в качестве web-сервера вполне вынесла. Вот он, в метре от меня, жужжит еле слышно.

Когда мы попадаем в другой город, в чужую квартиру, устраиваемся на другую работу, мы некоторое время чувствуем себя неуютно. Нет привычных вещей, обстановки, знакомых. Нет кнопки Пуск на рабочем столе. А что мне делать? А как я буду набирать реферат, доклад и т.д.? А как бродить по Интернету? Нас пугает все непривычное, незнакомое, и не потому, что мы не знаем, что делать – нам *неохота* вновь привыкать! Протянул руку за шкаф, даже в темноте – а там выключатель; машинально, почти автоматически, мы открываем и закрываем дверь на знакомый замок. Как вы думаете, в новой квартире мы с удивлением будем смотреть на пустую стенку и, не найдя на привычном месте выключателя, так и будем слоняться в темноте, или поищем место его расположения? Очевидно, второе, но с привычками пользователей операционных систем все не так просто.

Многие уже не помнят трудности перехода с DOS на Windows, хотя все еще сплошь и рядом работают окна с DOS'овскими программами. Кто на что смотрел, да так и не решился войти, вспомним русскую народную поговорку. Но! Всегда или надолго останутся те задачи, которые мы привыкли или вынуждены решать в Windows. За примером далеко ходить не надо: вот он – убийца легкого хлеба программистов середины 90-х, *«1С:Предприятие»* со всевозможными надстройками. Да-да, про *Wine* я знаю, но все ж согласитесь: когда срочно требуют «печатать расчетные листочки», тут не до экспериментов. В родной среде запустим *1С*, а потом, радуясь объявленной устной благодарности и (может быть!) премии, будем экспериментировать с *Wine* и постепенно переведем всю эту бухгалтерию на Linux.

Да и вообще, интересно провести скрещивание двух ОС. Тут нам на помощь придет свободная программа *VirtualBox*, запускающая Windows из-под Linux (и не только). Особенно сильное впечатление производят полноэкранный режим Windows и интегрированный рабочий стол двух ОС. А увидев сочетание умопомрачительного *Compiz Fusion* и *VirtualBox*, даже самые ортодоксальные любители Windows с придыханием выдавливают из себя: «У-у-х-ты!». LXF





Ответы

Есть вопрос по Open Source? Пишите нам по адресу: answers@linuxformat.ru!

1 30 ГБ не на месте

Я установил Linux Mint с LXF105, и, будучи новичком в Linux, последовал вашим инструкциям по разметке. Но я инсталлировал его на ноутбук с Windows, которая уже занимает два раздела, и разбив диск как было сказано в журнале, я обнаружил, что (так как разрешается иметь только четыре раздела или что-то в этом роде) не могу получить доступ к значительной части своего винчестера. Хотел выделить 10 ГБ на / и 512 МБ для подкачки, но редактор разделов Mint не дает мне ничего сделать.

Гэвин [Gavin]

Жесткий диск на PC может иметь лишь четыре раздела – одно из милых ограничений тех дней, когда люди считали себя вправе говорить «да кому надо больше, чем...» Решение проблемы – своего рода «костыль», но работает хорошо: один из этих четырех основных разделов помечается как расширенный, и он становится контейнером для так называемых логических разделов. Таким образом, можно иметь гораздо больше, чем четыре раздела – на сколько вам хватит места.

Проблема в том, что вы уже использовали свои четыре первичных раздела, и создавать расширенный раздел негде. Решение – удалить один из существующих разделов (лучше рядом с незанятым пространством) и использовать освободившееся место для логических дисков. Если ваши разделы идут в том порядке, в каком вы сказали – Windows, Windows, корневой раздел Linux, подкачка Linux, а затем неиспользуемое пространство – вам нужно только удалить раздел подкачки [swap], чтобы иметь возможность использовать все пространство, и вы можете сделать это даже на работающей системе.

Откройте терминал и выполните

```
sudo swapoff -a
```

чтобы отключить свой раздел подкачки; затем запустите редактор разделов, как вы делали ранее. Удалите раздел подкачки, и вы обнаружите, что теперь можно добавлять дополнительные разделы. Вам незачем беспокоиться о первичных/расширенных/логических нюансах: просто велите *GParted* создать раздел. Если он спросит, хотите ли вы получить первичный или логический раздел, ответьте «логический», и он позаботится о создании для него расширенного раздела. Первым делом нужно создать новый раздел подкачки взамен только что удаленного, и теперь он будет носить имя **sda5** – логические разделы всегда нумеруются числами от пяти, независимо от того, сколько у вас основных разделов.

Завершив редактирование разделов, включите новый раздел подкачки. Вернитесь в терминал и наберите

```
mkswap /dev/sda5
sudo gedit /etc/fstab
```

Найдите в редакторе строку, ссылающуюся на раздел подкачки, и замените часть UUID=xxxx на /dev/sda5 (или какое там имя получил ваш новый раздел подкачки). Когда вы перезагрузитесь, у вас снова будет раздел подкачки, а также 30 ГБ дискового пространства для экспериментов. **MC**

Это таинственный *gconf-editor*, где можно найти все скрытые настройки Gnome.

2 Скрытая панель

Я новичок в Linux, загрузивший Xubuntu из LXF99. После этого он был обновлен встроенными средствами, и экран *Grub* теперь говорит мне, что я использую 'ubuntu 7.10, kernel 2.6.22-14-generic'.

Проблема у меня в том, что я потерял панели задач и не могу больше находить и активировать приложения. Один раз это уже происходило, но тогда я использовал рабочий стол *Xfce*, так что щелчком правой кнопки мыши вызвал меню Applications и смог изменить настройки. А сейчас я сменил окружение на Gnome, и правый клик позволяет мне только доба- »

Наши эксперты

» Мы найдем эксперта по любому вопросу – от проблем с установкой системы или модемом до сетевого администрирования; главное – спросить!



Нейл Ботвик

Владелец ISP и экс-редактор дисков для нашего журнала, Нейл считает, что в Linux он от скуки на все руки.



Майк Сондерс

Майк был одним из создателей прототипа LXF – Linux Answers. Его специальности – программирование, оконные менеджеры, скрипты инициализации и SNES.



Пол Хадсон

Пол – местный супер-программист, и он может и хочет управиться со всеми вашими проблемами по части web и баз данных.



Грэм Моррисон

Когда он не обозревает кучи програмного обеспечения и не халтурит с MythTV, Грэм готов дать ответ касательно любого оборудования и проблем виртуализации.



Валентин Синицын

В редкие свободные минуты главный редактор нашего журнала обычно запускает *mcedit*, Чтобы отшлифовать какое-нибудь открытое приложение. Его любимая тема – настольный Linux.



Юлия Дронова

Когда ее компьютер не занят выполнением команды *emerge*, Юлия использует его, чтобы модерировать ЛинуксФорум.

КУДА ПОСЫЛАТЬ ВОПРОСЫ:

Пишите нам по адресу: answers@linuxformat.ru или спрашивайте на форуме: www.linuxforum.ru

» вить на рабочий стол еще один значок или выполнить несколько других бесполезных действий. Мне кажется, что если бы я мог запустить *Xfce* из терминала, то компьютер снова заработал бы.

Р.П.Трускотт [RP Truscott]

О Тут можно выделить два вопроса: как вернуться к *Xfce* и как восстановить панели Gnome. Ответ на первый прост – на экране входа в систему раскройте меню Сессия, расположенное под полями Имя пользователя/Пароль и выберите *Xfce* в качестве своего рабочего стола. Так вы сможете загрузить *Xfce*, а заодно установить его в качестве варианта по умолчанию.

Gnome не поставляется с Xubuntu, и в обновления он потом не включался, поэтому вы, вероятно, когда-то установили Gnome сами. Хотя панели можно удалять, Gnome, как правило, не позволяет убрать последнюю оставшуюся – то есть всегда должна быть одна рабочая панель. Дабы убедиться в том, что *gnome-panel* работает, и перезапустить его, если это необходимо, наберите **gnome-panel** в терминале. Тут должен появиться диалог ошибки, сообщающий, что панель уже запущена. Вы можете убить процесс панели – это заставит Gnome запустить новый:

```
killall gnome-panel
```

ваша панель может теперь появиться снова – если этого не произойдет, то она либо скрыта (невидима), либо имеет очень маленький размер. Чтобы изменить свойства вашей панели, даже если вы не видите ее, запустите **gconf-editor** из терминала. Перейдите к элементу Apps > Panel > toplevels > panel_0. Отсюда можно изменить многие параметры панели, хотя бы и невидимой. Нажмите на пункт в колонке Name, чтобы увидеть описание настройки; нажмите на колонку Value, чтобы ее изменить.

Получив панель обратно, вы можете обнаружить, что она пуста, и в этом случае придется восстановить необходимые вам апплеты. Щелкните правой кнопкой мыши на том месте на панели, где вы хотите добавить апплет, и выберите Add to panel… Чтобы добавить меню приложений, выберите Menu Bar (не Main Menu) из нижней части списка. **НБ**

3 LVM

В Новая инсталяция Fedora 8 выглядит из моей стандартной установки Fedora Core 5 как два раздела: **/dev/hda7** и 8, как в fdisk, так и в **/dev**, безо всяких там **/dev/VolumeXX/GroupXX**, как пишут в Интернете. Мне хотелось бы добавить запись для Fedora 8 в мой файл **grub.conf**, чтобы я мог загружать ее. Я успешно пользуюсь Fedora Core 5 уже давно, и она содержит все мои настройки и данные.

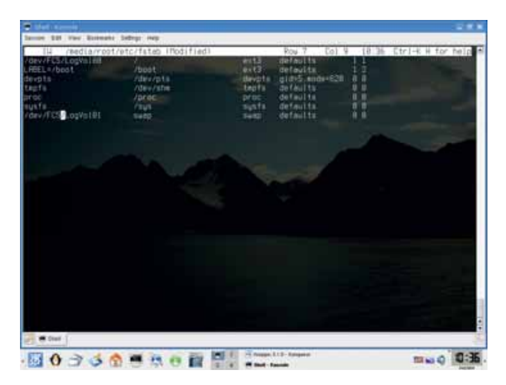
Майкл А. Вард [Michael A Ward]

О Fedora Core 5 и Fedora 8 используют похожие настройки LVM, и это является источником вашей проблемы. Оба дистрибутива дают первой группе томов название **VolGroup00**. Поскольку LVM различает группы томов по именам, это сбивает его с толку. В результате одна из групп игнорируется, потому-то вы и не можете увидеть ее содержимое. Решение – изменить одно из имен, но вы не сможете проделать это с Core 5, так как ее группа используется, и не сможете переименовать группу Fedora 8 – система не увидит ее, пока вы не смените имя. Наилучший способ исправить ситуацию – загрузиться с LiveCD вроде Knoppix, чтобы беспрепятственно добраться до групп томов. Когда Knoppix запустится, откройте терминал и наберите **su** (пароль не требуется), чтобы стать root.

Последние версии инструментов LVM умеют определять различные группы томов с одинаковыми названиями. Запустите *vgdisplay*, и вы увидите две группы с именами VolGroup00, но различными UUID. Вы, вероятно, сумеете определить принадлежность группы по ее размеру, поэтому переименуйте ту, что относится к Fedora 8, командой

```
vgrename xxxxxxxxx Fedora8
```

заменив xxxxxxxxx на UUID из вывода **vgdisplay**. Вы могли бы сейчас перезагрузиться в Core 5 и увидеть



› Переименовав с помощью Knoppix группу логических томов, для снятия конфликта, отредактируйте /etc/fstab, чтобы ваше ядро смогло найти ОС.

оба набора логических томов, хотя вам пришлось бы отредактировать **/etc/fstab** в корневой файловой системе Fedora 8 в соответствии с новым названием. Однако добавив еще один диск с Fedora, вы столкнулись бы с той же самой проблемой, так что лучше переименуйте оба тома прямо сейчас; после этого, чтобы загрузиться с измененными названиями групп, вам придется кое-что наладить. Во-первых, измените меню *Grub* с помощью команд

```
mount /dev/hda7
joe /media/hda7/grub/menu.lst
```

монтирующих загрузочный раздел и открывающих файл меню в редакторе. Исправьте строки с VolGroup00 и нажмите Ctrl + K, X для выхода с сохранением. Теперь смонтируйте корневую файловую систему и отредактируйте **/etc/fstab** так:

```
vgchange -a y
mkdir /media/root
mount /dev/FC5/LogVol00 /media/root
joe /media/root/etc/fstab
```

Первая строка активизирует переименованные группы томов. Измените два вхождения VolGroup00 на FC5 или на то имя, которое вы выбрали для Core 5. Нажмите Ctrl + K, X, чтобы сохранить файл и перезагрузите компьютер. Естественно, если вам лень это делать, можно просто переустановить Fedora 8 и задать другое название группы томов во время инсталляции.

Теперь, когда конфликт с именами групп разрешен, можно добавить в загрузчик Core 5 запись о Fedora 8:

```
title Fedora 8
root (hd1,4)
chainloader +1
```

передаст контроль загрузчику на **/dev/hdb5**. Рекомендую также создать отдельный раздел /home, с тем, чтобы при будущих обновлениях не терять свои личные данные и настройки. **ПХ**



Вопрос-победитель (русская версия)

Алексей получает подарочный сертификат на 1000 рублей от интернет-магазина ГНУ/Линуксцентр! Просим победителя выйти на связь с редакцией: info@linuxformat.ru

Миграция по всем правилам

В Руководство фирмы, где я работаю системным администратором, рассматривает возможность перехода на свободное ПО. Технических вопросов почти не возникает, но я не знаю, как провести это через бухгалтерию, и какие документы могут затребовать при проверке компетентные органы.

Алексей

О Проверяющие будут обращать внимание на две вещи:

1 Что на все программное обеспечение, которое будет найдено на ваших компьютерах и серверах, имеются соответствующие лицензии;

2 Что все программное обеспечение принято к учету в бухгалтерии. Для этого вашему бухгалтеру потребуются первичные документы (счет, счет-фактура и товарная накладная либо акт выполненных работ).

Не все сотрудники компетентных органов знакомы с понятиями «свободное ПО», «Linux», «GPL» и прочими, так что бумажное подтверждение ваших прав на использование программ (что бы там ни говорил Ричард Столлмен про наши законы) будет не лишним – такую лицензию сегодня можно приобрести во многих интернет-магазинах; она также вкладывается в коробки с коммерческими дистрибутивами Linux вроде Mandriva Powerpack. Кроме того, имейте в виду, что если компьютеры «прошли» по бухгалтерии в 2005 году, а ПО к ним – лишь в 2008, это также может вызвать вопросы. **Павел Фролов**

4 Debian без Интернета

В Я – новичок в Linux, и в настоящее время изучаю основы использования Debian-систем, но у меня есть проблемы с установкой новых программ. Дома у меня нет доступа к Интернету, и я пользуюсь компьютерами в местной общественной библиотеке (которые работают под Windows) и переношу данные в мой компьютер на USB-брелке. Я попытался установить загруженные мной программы Linux с помощью менеджера пакетов, но это не

сработало. При инсталляции программ с DVD-дисков, которые идут, например, с вашим журналом, возникает та же проблема. Я просмотрел книги про Linux и прошлые ответы на вопросы в журнале о проблемах, похожих на мою, но не смог найти ничего, что могло бы мне помочь. Подскажите, пожалуйста, как выполнить установку программы, сохраненной на брелке, или с DVD?

Джолион Пэриш [Jolyon Parish]

О Менеджер пакетов *Dpkg* в Debian, который является низкоуровневым фундаментом для графических программ-менеджеров типа *Synaptic*, способен выполнить установку прямо из файла пакета. Для этого запустите (в терминале) команду:

```
sudo dpkg --install somepackage-1.2.3.deb
```

если вы используете Ubuntu или одну из его производных, или

```
su
dpkg --install somepackage-1.2.3.deb
```

для прочих дистрибутивов на базе Debian. Можно установить несколько пакетов сразу, либо путем перечисления их имен в командной строке, либо передав программе имя каталога, их содержащего:

```
sudo dpkg --install --recursive /media/usbstick
```

Однако сперва вы должны узнать, какие файлы необходимо загрузить, так что есть еще один вариант. Запустите *Synaptic*, отметьте пакеты, которые вы хотели бы установить, и используйте пункт меню File > Generate Package Download Script. Он создает скрипт для скачивания необходимых вам файлов. Хотя использовать его напрямую на большинстве компьютеров с Windows не получится, вы можете скопировать и вставить URL-адреса из этого сценария в вашу программу закачки и разместить загруженные файлы на USB-брелке. Затем подключите его обратно к своему домашнему компьютеру, запустите *Synaptic* и выберите File > Add Downloaded Packages; перейдите в каталог, содержащий скачанные вами файлы, нажмите кнопку Open, и *Synaptic* установит их для вас.

Обновить список доступных программ немного сложнее, но это тоже возможно. Загляните в **/etc/apt/sources.list**. Для каждого источника вы увидите в нем строку вроде этой:

```
deb http://gb.archive.ubuntu.com/ubuntu/ gutsy
main restricted
```

В данном примере, откройте http://gb.archive.ubuntu.com/ubuntu/dists/gutsy и перейдите в директории Main и Restricted. В каждой из них вы найдете каталог **binary-i386**; скачайте файлы **Packages.bz2** и **Release** и сохраните их, включив в имя полный путь. Символы / заменяются на _, вот так:

```
gb.archive.ubuntu.com_ubuntu_dists_gutsy_
binary-i386_Packages.bz2
```

Скопируйте каждый файл на ваш USB-брелок. Возьмите его домой, перенесите все файлы в **/var/lib/apt/lists/** от имени root и распакуйте файлы .bz2. Самый легкий способ сделать это – из терминала, с помощью

```
cd /var/lib/apt/lists/
sudo cp /media/usbstick/*.bz2 .
sudo bunzip2 *.bz2
```

Теперь запустите *Synaptic*, и он должен будет иметь все последние версии. Это немного муторно, но система изначально была предназначена для использования с интернет-соединением. **НБ**

5 Тест памяти

В Некоторые из дисков LXFDVD и дистрибутивов Linux предлагают загрузочное меню с различными вариантами установки, включая (иногда) утилиту проверки оборудования, такую как *Memtest*. Можно ли получить ее код и загружать программу с жесткого диска, и как это сделать?

Formatix, с форумов



› Заставить *Memtest86* запускаться из загрузочного меню не сложнее, чем скопировать один файл и отредактировать другой.

О Это не только можно, но и очень просто. Некоторые дистрибутивы имеют готовые пакеты для *Memtest86* или *Memtest86+*, ответвления от первоначального проекта. В таких случаях установка из менеджера пакетов, как правило, добавит пункт в меню загрузчика и сделает все необходимое за вас. Если вы хотите установить программу вручную, зайдите на www.memtest86.com или www.memtest.org, в зависимости от варианта, который вы хотите попробовать, и скачайте предварительно скомпилированную версию. В случае *Memtest86*, она будет помечена как «устанавливается в Windows или DOS», однако она годится и для Linux. Распакуйте архив и скопируйте **.bin**-файл в **/boot** – для этого вам нужно быть суперпользователем. Затем, также от имени root, отредактируйте **/boot/grub/menu.lst** (некоторые дис- »



Вопрос-победитель (английская версия)

«Горячая» замена SATA

В У меня есть несколько 500-ГБ жестких дисков SATA со всеми моими фильмами. Вместо того, чтобы разместить их на сервере и получать доступ к ним по проводу, я решил установить на моем медиа-компьютере мобильное шасси. Единственная проблема заключается в том, что я должен выключать компьютер, чтобы менять жесткие диски.

Хочу иметь возможность «горячей» замены дисков, но не представляю, как сделать это под Linux, когда таблица дисков загружается при старте ядра.

Rod

О Вам поможет «горячее подключение» SATA, но оно зависит от оборудования сразу в двух местах. Используемое Вами шасси должно поддерживать «горячую» замену; большинство из них ее поддерживают, но на всякий случай проверьте это перед покупкой. Также необходима блокировка, хотя некоторые шасси используют задвижные защелки, а не замки: они не только фиксируют накопитель на месте, но также контролируют подаваемое к диску питание. Разблокирование привода обесточит диск, чтобы он не вращался, когда вы физически его достаете.

Во-вторых, «горячую» замену должен поддерживать ваш SATA-контроллер. Он должен уметь распознать, что диск был отключен или подключен, и сообщить эту информацию ядру. Наконец, операционная система должна поддерживать «горячую» замену SATA-приводов, подобно устройствам USB или FireWire.

Выявление подходящих контроллеров не так уж просто. Я был вполне удовлетворен контроллером Intel ICH8, работающим в режиме AHCI, что, по-видимому, и является важнейшим фактором. Если ваши SATA-контроллеры совместимы с AHCI (когда это так, в BIOS часто имеется опция для ее включения или отключения), то все должно быть OK, но сперва поищите в Google данные для вашего контроллера.

Посмотрите журнал системы с помощью команды

```
tail -f /var/log/messages
```

во время установки и замены накопителей. Вы должны увидеть всякие сообщения насчет исчезновения и появления дисков. Если это так, то вы готовы ими воспользоваться, хотя есть еще один фактор, который Вам, возможно, придется учесть. Если вы хотите, чтобы диски монтировались автоматически, и ваша система использует для этого pmount (утилиту, позволяющую монтировать диски от имени обычного пользователя без записи в **/etc/fstab**), Вам, возможно, придется отредактировать **/etc/pmount.allow**. Если диски рассматриваются как несъемные, как обычно происходит с винчестерами SATA, то pmount откажется монтировать их, пока вы не добавите имя устройства в **/etc/pmount.allow**, например, так:

```
echo '/dev/sdb1' >>/etc/pmount. allow
echo '/dev/sdc[123]' >>/etc/
pmount.allow
echo '/dev/sdd*' >>/etc/pmount. allow
```

Первая строка позволяет одному конкретному разделу быть смонтированным pmount, вторая разрешает три конкретных раздела на жестком диске, а третья позволяет делать это с любым разделом на sdd. Одиночные кавычки использованы, чтобы запретить оболочке интерпретировать маски. **НБ**

» трибутивы используют **/boot/grub/grub.conf**) и добавьте одну или обе следующих записи, в зависимости от варианта, который вы установили.

```
title memtest86
 kernel /boot/memtest.bin
 title memtest86+
 kernel /boot/memtest86+-2.01.bin
```

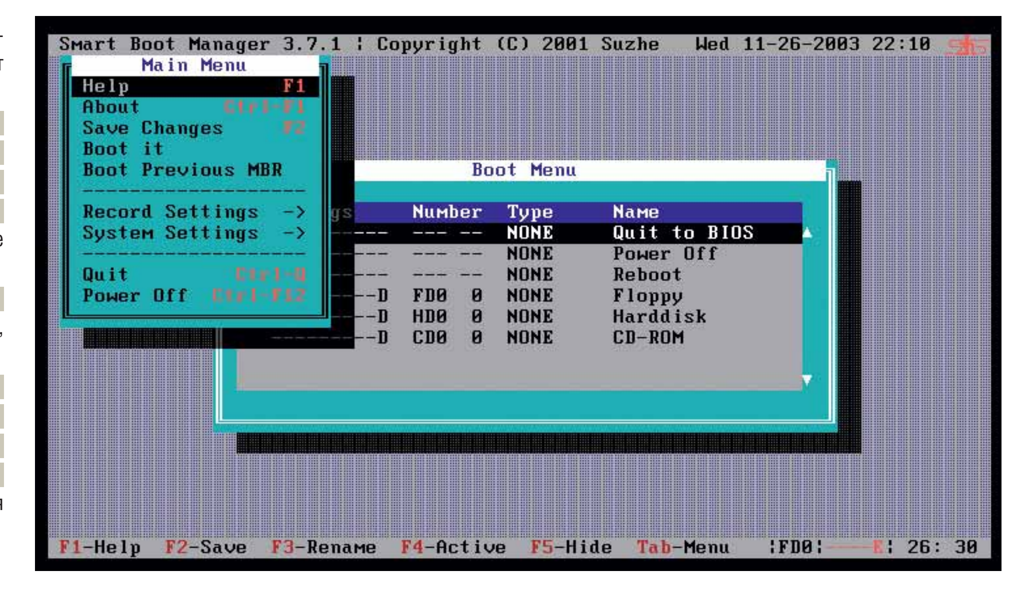
Если у вас отдельный загрузочный раздел, можете опустить **/boot**, вот так:

```
 kernel /memtest86+-2.01.bin
```

Если ваш дистрибутив использует *Lilo* вместо *Grub*, добавьте в **/etc/lilo.conf** следующее:

```
 image=/boot/memtest.bin
 label=memtest86
 image=/boot/memtest86+-2.01.bin
 label=memtest86+
```

Не забудьте запустить **/sbin/lilo** после изменения lilo.conf! **MC**



› Smart Boot Manager может убедить загружаться некоторые «упрямые» CD- или DVD-диски.

6 Перезапуск Ubuntu

В У меня установлен Feisty Fawn, и он неплохо работает на старом ПК. Я не имел проблем с большей частью программ, кроме случаев, когда пытался поиграть. Единственным опечалившим меня приложением была карточная игра, с которой я долго развлекался еще на XP и которая использует два банка из четырех карт. При каждой моей попытке в нее играть она на каком-то этапе вешает компьютер. Я думал, что все будет в порядке, если я перегружу Feisty, но я не могу попробовать любой другой Live-диск, так как не знаю, как выйти из Ubuntu. Как мне вернуться к чистому жесткому диску, чтобы начать все сначала? Я не умею выполнять удаление, и при работе с Windows у меня такой проблемы не было.

Еще я не смог подключиться к Интернету, и у меня уже скопилось много файлов, которые нуждаются в связи, прежде чем они смогут быть использованы, и я не хочу постоянно перебрасывать мое широкополосное подключение между двумя компьютерами. Я, возможно, делаю что-то неправильно, но я считаю, что должен добиться результата.

Дэнни Грин [Danny Green]

О Переустановка всей операционной системы для решения проблем одной карточной игры (уж извините) не есть хорошее решение. Вы потратите час и вернетесь на старое место. Правильный ответ заключается в поиске решения в Google, форумах Ubuntu или в размещении вопроса с подробной информацией об игре на ресурсе вроде www.linuxforum.ru. Если эта игра была включена в Ubuntu, исправить ее, скорее всего, будет довольно просто.

Также, похоже, ваш компьютер настроен на загрузку с жесткого диска перед CD/DVD, поэтому вы должны в явном виде приказать ему запускаться с Live CD. Большинство компьютерных BIOS имеют загрузочное меню – нажмите определенную клавишу в момент старта системы, и компьютер спросит вас, с какого диска вы хотите грузиться. Если его нет, войдите в BIOS при включении питания компьютера и измените порядок загрузки, поставив CD/DVD-привод



Часто задаваемые вопросы...

Разделы и файловые системы

Зачем нарезать ваш жесткий диск на куски, как пирог?

» Зачем надо иметь так много разделов и файловых систем в установке Linux?
Жесткий диск разбит на разделы – части, которые действуют, как если бы они были самостоятельными накопителями. Каждый раздел затем использует файловую систему для хранения данных.

» А почему в Linux их больше одного? В Windows столько нет!
Хранение различных типов данных по отдельности может повысить производительность и безопасность. Хранение данных подкачки в отдельной файловой системе более эффективно, чем запись их в файл в корневом разделе. Вредные процессы могут испортить подкачку, но больше они ничего не сломают.

» Значит, мне хватит раздела подкачки и корневого раздела, как в Ubuntu?
Этого достаточно, но такая схема не оптимальна. По крайней мере, для настольных систем (в том числе ноутбуков) следует использовать отдельный раздел для домашних директорий (**/home**). Тогда ваши персональные настройки и данные будут содержаться отдельно от файлов ОС.

» Зачем мне это надо?
Когда-нибудь вам захочется попробовать другие дистрибутивы, либо обновить систему. Если все данные хранятся в корневом разделе, они будут стерты при инсталляции, и вам придется создать резервные копии всех ваших настроек и файлов на внешних носителях, прежде чем вы сможете что-нибудь установить. При отдельном разделе /home правильный инсталлятор оставит данные в неповрежденном состоянии. После установки вы можете загрузить машину и начать использовать ее так же, как вы делали в старой системе.

» Надо ли иметь еще какие-либо разделы?
Для сервера хорошей идеей будет иметь отдельную файловую систему для **/var**. Это также полезно и для настольных компьютеров: если запускающиеся процессы переполнят системные журналы, хранящиеся в **/var/log**, они не смогут отобрать дисковое пространство, занимаемое файлами ОС.

» А если я захочу синхронизировать каталоги на моих Linux- и Windows-компьютерах?
Rsync доступен для Windows, но не применяйте опцию **--archive**, если вы используете FAT-разделы, потому что FAT не поддерживает права доступа к файлам. Смотрите man-страницу *Rsync* для списка опций.

» Столько типов файловых систем... зачем нам ext2, ext3, ReiserFS, XFS и другие?
Каждая из них имеет свои сильные и слабые стороны. Ext2 – первоначальная файловая система Linux, и она очень быстра, но не очень хорошо показывает себя при внезапных сбоях. Ext3 – журналируемая файловая система, а это значит, что она надежнее и быстрее восстанавливается в случае аварий за счет небольшой потери производительности. ReiserFS особенно хороша при работе с небольшими файлами, из-за эффективного использования дискового пространства и скорости. XFS является моим фаворитом. Она быстра, особенно при работе с большими файлами, но плохо обрабатывает отключения электропитания, и лучше всего использовать ее на чем-то с резервной батареей – ноутбуке или системе, защищенной ИБП.

перед жестким диском. Третий вариант – использовать *Smart Boot Manager*, который можно найти на любом из наших LXFDVD. Скопируйте его на дискету, открыв терминал и запустив

```
sudo cat /media/cdrom0/Essentials/SBM/
sbootmgr.dsk >/dev/fd0
```

с дискетой в дисководе. Теперь перегрузитесь и выберите 'CDROM' из меню *Smart Boot Manager*. Если вы используете DVD, не волнуйтесь – выбирайте также CDROM.

Вы не сказали, какой у вас тип широкополосного соединения, но если ваш модем имеет порт Ethernet, вы должны иметь возможность подключить его напрямую к компьютеру через стандартный сетевой кабель, а затем настроить Ubuntu на получение адреса автоматически (это стоит по умолчанию, и менять ничего не нужно). Чтобы выходить в Интернет с двух компьютеров сразу, понадобится маршрутизатор. Он подключается либо к кабельному модему, либо (если у вас ADSL) объединен с ним. В последнем случае надо просто подсоединить устройство к телефонной линии и «воткнуть» в него все ваши ПК. В наши дни маршрутизаторы стоят дешево, и подключить компьютеры к Интернету и друг к другу очень просто. **НБ**

7 Загрузка с внешнего диска

Я новичок в Linux и изучаю все дистрибутивы. У меня была двойная загрузка XP с Ubuntu, а потом я добавил второй диск и водрузил на него SUSE 10.3. Ubuntu при установке создал загрузочное меню, включающее Windows, а когда я инсталлировал SUSE, он сделал собственное меню, содержащее и Windows, и Ubuntu. При выборе Ubuntu из меню SUSE я возвращаюсь в меню Ubuntu, которое работает как раньше.

А теперь я хочу загрузиться с 250-ГБ USB-диска с PCLinuxOS. Когда я попытался инсталлировать PCLinuxOS напрямую, он установил свое загрузочное меню, удалив меню SUSE и оставив только себя и Windows. Есть ли простой способ добавить операционную систему на внешнем диске в меню SUSE или быстро создать новое меню, включающее все ОС, и на внутренних, и на внешних дисках? Я не сильно разбираюсь в командной строке, так как мигрировал из Windows недавно. Почитав всякие учебники по *Grub*, я разобрался в терминах «раздел», «метка» и «нумерация», но не хотелось бы зря угробить существующую систему загрузки. Мой второй диск распознается BIOS.

Питер Андерхилл [Peter Underhill]

Обычно дистрибутивы устанавливают загрузчик в главную загрузочную запись (MBR) первого жесткого диска. Когда компьютер запускается, он видит здесь загрузчик и передает ему управление. Проблема состоит в том, что каждая новая установка перезаписывает загрузчик предыдущей, как вы уже выяснили. Если у вас есть работающее загрузочное меню, которое вы хотите оставить, можно предотвратить его перезапись любым другим дистрибутивом, установив загрузчик последнего в корневой раздел (данная опция может быть скрыта в секции Advanced инсталлятора). Это означает, что загрузчик дистрибутива не записывается в MBR и не затирает оригинальный загрузчик – вместо этого он ставится в начало основного системного раздела его собственного дистрибутива. Теперь вы можете изменить первоначальное загрузочное меню, чтобы добавить пункт, передающий контроль новому дистрибутиву.

Сначала вам нужно выяснить, где находится корневой раздел – вы должны были видеть это во время установки. Если вы устанавливаете систему в первый раздел внешнего диска, то, возможно, это **/dev/sdb1**. Linux называет жесткие диски **sd** (а иногда **hd**) и следующей за этим буквой – '**a**' для первого диска и так далее – и нумерует разделы, начиная с единицы. Для вящей путаницы, *Grub* использует иную схему и помечает диски как **(hdx,y)**, где **x** – это номер привода, а **y** – номер раздела на этом диске, начиная с нуля. Так что первый раздел на втором диске называется **/dev/sdb1** в терминах Linux и **(hd1,0)** в *Grub*.

Теперь вы понимаете, что надо загрузить дистрибутив-владелец вашего основного загрузчика, в данном случае – SUSE, и внести изменения, добавив запись для нового дистрибутива. Вы можете изменить загрузчик в *Yast* или напрямую, отредактировав в терминале файл **/boot/grub/menu.lst** (это следует делать от имени root)

```
su
nano /boot/grub/menu.lst
```

Прокрутите текст вниз (вы увидите существующие пункты меню), а затем добавьте еще один для вашего нового дистрибутива:

```
title My New Distro
root (hd1,0)
chainloader +1
```

Первая строка – это текст пункта меню; вторая сообщает загрузчику местоположение раздела, содержащего новый дистрибутив, а последняя передает контроль загрузчику нового дистрибутива. Выберите ее, и вы увидите второе загрузочное меню. Если *Grub* сообщит об ошибке 21 или 22, значит, вы неправильно указали раздел в строке root. Вы можете выделить новый пункт меню и нажать E, выбрать строку root и, нажав E снова, изменить его. Подправив строку root, нажмите Esc для применения изменений и B для загрузки. Повредить систему этими экспериментами вы не сможете. Найдя правильное значение, отредактируйте файл **menu.lst**, как было указано выше, сделав изменения постоянными. **НБ**

8 Mandriva заартачилась

На днях одногруппник Федя принес мне Mandriva 2008 Spring. Проблема в том, что диск оказался незагрузочным. Я где-то читал, что это можно исправить, используя программу *mkisofs* и каталог **isolinux**. Подскажите пожалуйста, как именно это сделать.

Андрей Кондратьев

Тому, что диск не загружается, может быть несколько причин:

1 Дефект носителя

2 Испорченный образ (MD5-сумма скачанного ISO-файла не соответствует эталонной

3 Определенное сочетание аппаратных устройств, индивидуальное для каждого »



Краткая справка про...

top

Казалось ли вам когда-нибудь, что ваш компьютер еле ползает, не реагирует на ваши действия или вгоняет в ступор жесткий диск? Если такое случится опять, наберите в терминале *top*. Эта программа вмещает много информации в малую площадь. По умолчанию в верхней строке отображаются: время работы системы (uptime), число пользователей, количество выполняющихся процессов и т.д. Здесь также выводится использование процессорного времени для пользовательских задач (us), системных операций (sy), неприоритетных задач (ni), бездействующих (id) и ожидающих (wa) процессов. Бездействие показывает, какой запас процессорных ресурсов у вас еще есть. Ожидающие процессы – это такие программы, которые ждут каких-то событий (обычно ввода/вывода). Ниже показывается список задач, отсортированный по потреблению процессорного времени – тут легко определить, кто сожрал львиную долю циклов вашего CPU.

Top не просто отображает процессы: вы можете влиять на их работу. Нажмите K, чтобы убить процесс, или R – чтобы понизить его приоритет, если он отбирает слишком много ресурсов. Понизить приоритет – все равно, что сказать процессу «будь повежливей, пропускай других вперед». Убийство и понижение приоритета требуют указания идентификатора процесса (PID), помещенного в левой колонке. Чтобы нечаянно не угробить невинного, после набора **top** в *X*-терминале просто вырежьте и скопируйте нужный ID.

У *top* есть множество опций, большинство из которых имеют горячие клавиши. Самая нужная – H, она выводит справочную информацию.

› Top показывает, сколько используется памяти и циклов процессора и какие программы доступны.

» компьютера, но в сумме делающее загрузку ядра невозможной по причине отсутствия каких-либо модулей (драйверов)

4 «Нечитаемость» загрузочного диска, в частности, Mandriva испытывает кое-какие трудности с некоторыми CD/DVD-приводами NEC.

5 Возможны проблемы с RAID-массивами, опять же из-за драйверов

Isolinux – это загрузчик для Linux/i386, который работает с компакт-дисками формата ISO 9660/El Torito в режиме “no emulation” (без эмуляции), т.е. когда загрузочная информация хранится непосредственно на CD. Это позволяет избежать создания образа для эмуляции загрузки с дискеты или винчестера. *mkisofs* – утилита командной строки из пакета *cdrtools*, превращающая переданный ей в качестве параметра каталог в файловую систему (ISO-образ), пригодную для записи на CD- или DVD-диск.

Чтобы создать загружаемый CD с помощью *Isolinux*, скопируйте директорию **isolinux/** с CD-ROM вашей Mandriva Linux 2008.0 Spring во временный каталог при помощи команды:

```
cp -r <путь_до_CD-ROM>/isolinux/ <временный_каталог>
```

Перейдите во временный_каталог:

```
cd <временный_каталог>
```

Убедитесь в том, что скопированные вами файлы обладают требуемыми правами:

```
chmod u+w isolinux/*
```

Наконец, используйте следующую команду для создания файла ISO-образа:

```
mkisofs -o file.iso -b isolinux.bin -c boot.cat -no-emul-boot
-boot-load-size 4 -boot-info-table -R -J -v -T isolinux/
```

ЮД

9 Ночной дозор

В Я хочу поставить закачку на ночь. Как я понимаю, для этого следует использовать две программы: *at* («будильник») и *wget* (собственно «качалка»). Последняя вопросов не вызывает (прекрасная утилита!), а вот *at* не работает. Пишу:

```
ubuntu@ubuntu:~$ at 06:32
warning: commands will be executed using /bin/sh
at> wget http://<здесь ссылка на объект скачивания>
at> <EOT>
job 1 at Mon Jun 16 06:32:00 2008
ubuntu@ubuntu:~$
```

но ничего не происходит. Пробовал с другими командами (*Nautilus*, *Gedit*) вместо *wget* – результат тот же. В Сети предлагают следующее

```
echo export DISPLAY=$DISPLAY \&\& gedit |at 06:32
```

но эта команда мне не совсем понятна (точнее, совсем не понятна), и потом в man-странице *at* про нее ничего не говорится.

Владимир Клопцов

О Переменная **DISPLAY** здесь не поможет – она влияет только на графические приложения (к таковым относятся перечисленные вами *Nautilus* и *Gedit*) и показывает, какой *X*-дисплей им следует использовать. Корень вашей проблемы, скорее всего, кроется в других переменных окружения. Задания *at* выполняются не в вашей стандартной оболочке, а в своей собственной, поэтому не стоит предполагать, что переменная **PATH** (она содержит пути, где происходит поиск исполняемых программ) имеет то же самое значение, что и в вашем терминале. Укажите в задании полный путь до *wget*, скажем **/usr/bin/wget** (его можно получить в окне консоли командой **which wget**). Кроме этого, удостоверьтесь, что демон atd *запущен* и работает нормально. **ВС**

Нужна помощь!

› Для наилучшего ответа на ваш вопрос нам нужно знать как можно больше подробностей. Детально опишите конфигурацию системы. Если вы получили сообщение об ошибке, приведите текст сообщения и точно опишите вызвавшие его действия. Если у вас проблемы с оборудованием, то опишите его. Если Linux уже запущен, то выполните в root-терминале следующие команды и прикрепите к письму файл **system.txt**:

```
uname -a >system.txt
lspci >>system.txt
lspci -vv >>system.txt
```

› Пожалуйста, помните, что сотрудники журнала НЕ являются авторами или разработчиками Linux, любых пакетов или дистрибутивов. Зачастую люди, отвечающие за приложения, выкладывают большую часть информации на web-сайты. Попробуйте почитать документацию!

Мы стараемся ответить на все вопросы. Если вы не нашли ответ на свой, это, возможно, потому, что мы уже ответили на похожий вопрос.

Большой вопрос Как обмениваться файлами между Linux- и Windows-машинами?

10 Samba, крошка!

В На моей настольной Linux-машине запущена *Samba*, чтобы мой ноутбук с Windows мог работать с файлами на этом компьютере. Если мне нужно получить доступ к файлам на моей Linux-машине с ноутбука опять же с Linux, нужно ли мне запускать *Samba*? Я однажды попытался использовать *TightVNC*, и смог увидеть чужой рабочий стол. Могу ли я использовать его для передачи файлов, как в *Samba*? Я также слышал о штуке под названием KDE Remote Desktop Connection. Это что-то вроде VNC? И какой метод предпочтительнее – VNC или Remote Desktop?

Я также читал про SSH, но еще его не пробовал. В Википедии говорится, что «Secure Shell или SSH – это сетевой протокол, позволяющий обмениваться данными, используя защищенный канал между двумя компьютерами.» Это что-то вроде *Samba*, которая позволяет связывать мой ноутбук и настольный компьютер, или обмен можно производить только между компьютерами с Linux? Или это вообще другой зверь?

Мик [Mic]

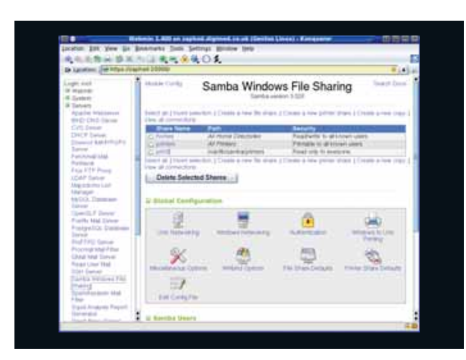
О *Samba*-сервер позволяет компьютерам получать доступ к файлам с использованием протоколов Windows: SMB и CIFS. Хотя *Samba* вначале предназначалась для работы с файлами не-Windows компьютеров в компьютерной сети Windows, она уже переросла данную стадию. Так как это сервер, необязательно запускать его на Linux-компьютере, с которого вы получаете доступ к файлам Windows-системы: здесь хватит клиентского ПО. Оно, как правило, устанавливается по умолчанию и часто включается в отдельный пакет, чтобы не устанавливать всю *Samba* только для работы с файлами в Windows-сети.

Samba можно использовать и для обмена файлами между Linux-компьютерами. Существует также более родной метод, который называется NFS (Network File System, сетевая файловая система), но если ваша сеть содержит смесь Windows и других ОС, то проще придерживаться *Samba* для всего.

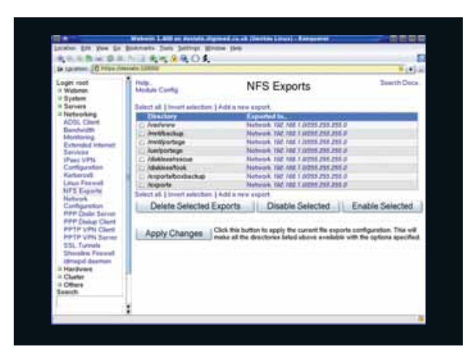
TightVNC – дело другое: это вариант первоначального пакета VNC, позволяющий получить удаленный доступ к графическому рабочему столу другого компьютера. Он не включает средства передачи файлов, так как вы делаете все на удаленном компьютере, но с использованием локальной клавиатуры, мыши и монитора. *Remote Desktop Connection* в KDE является интерфейсом к VNC и Windows Remote Desktop Protocol. Он может подключаться к компьютерам с использованием любого метода, определяя лучший протокол для каждого конкретного соединения.

SSH – третий вариант, обеспечивающий способ входа в командную оболочку удаленного компьютера с помощью шифрованного канала, делая ее безопасной для административных задач при незащищенном соединении, таком как Интернет. SSH также предоставляет средства передачи файлов через командную строку и программы *scp* и *sftp*. Последняя может быть также использована с графическими файл-менеджерами. Наберите **sftp://пользователь@домен/путь/к/директории** в адресной строке *Konqueror* для отображения содержимого каталога на удаленном компьютере (конечно, если у вас есть SSH-доступ к нему). Существует также SSH-программа для Windows, называемая *Putty* (www.chiark.greenend.org.uk/~sgtatham /putty). **НБ** LXF

Варианты удаленного входа для Linux и Windows



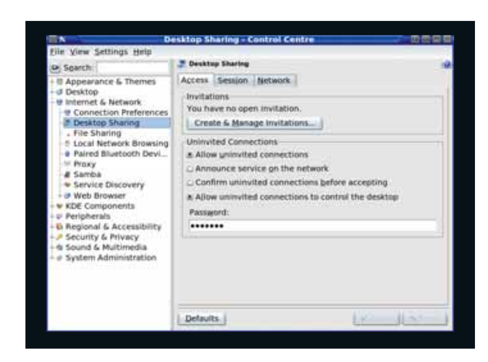
Samba
Samba делает файлы и принтеры доступными для Windows-компьютеров в вашей сети, но может использоваться и в связке Linux-c-Linux.



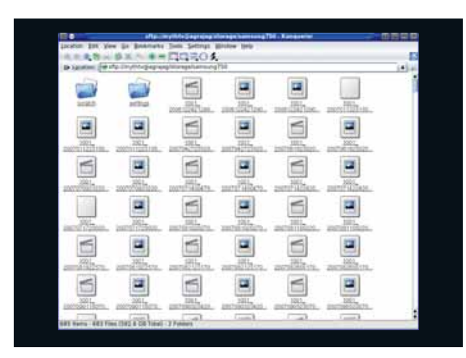
Альтернатива *nix
NFS (Network File System), разработанная Sun, используется для обмена файлами во всех Unix-подобных ОС, включая Linux и OS X.



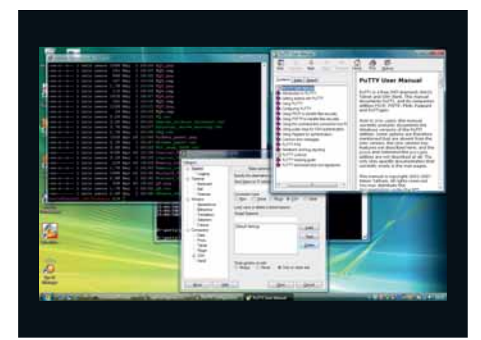
Удаленный рабочий стол
И VNC, и *TightVNC*, и Remote Desktop Protocol обеспечивают удаленный доступ к другому компьютеру. Это не системы передачи файлов.



Решение от KDE
Утилита удаленного доступа в KDE использует VNC или *Remote Desktop*, чтобы показать ваш рабочий стол на другом компьютере.



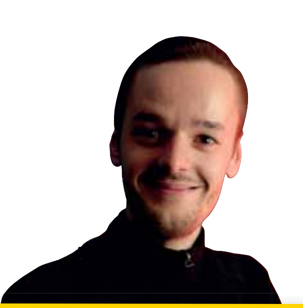
SSH
SSH также предоставляет средства для просмотра и копирования файлов с удаленного компьютера через безопасное зашифрованное соединение.



/home из дому
Putty – пакет SSH для Windows, он позволяет получить доступ к вашей Linux-машине даже из свободной от пингвинов зоны.

LXF DVD108

Шляпы долой – выбираем новую красную шапочку от Linux.



Запустив DVD, вы, очевидно, заметите, что в этом месяце раздел **Главное** практически пуст.

Чтобы вместить новый релиз Fedora и еще много всего хорошего, о чем пишет наш журнал, нам пришлось отправить большую часть **Главного** в заслуженный отпуск. Но, отдохнув, оно снова вернется к нам в следующем месяце с новыми видеодрайверами и релизами ядра для всех желающих удерживать свои Linux-машины на передовой последних нововведений..

Майк Сондерс
Редактор диска
mike.saunders@futurenet.co.uk

Дистрибутив Linux

Fedora 9

Fedora – восхитительный дистрибутив. Спонсируемый Red Hat и подкрепленный сообществом, он постоянно удерживается на тонкой грани между суперновизной и стабильностью рабочей станции. Если он слишком далеко заберется в сторону программ бета-уровня, то перестанет отличаться надежностью в повседневном применении; а если он не будет идти в ногу с самым современным ПО, его затмят другие дистрибутивы. Поэтому команда Fedora сохраняет баланс, предлагая самые современные программы и не допуская нестабильности. Fedora 9 имеет ряд улучшений, о которых вы можете прочитать в нашем обзоре на стр. 8 – см. также видео в разделе **Журнал** на **LXFDVD**.

Наш DVD включает полный релиз Fedora 9, готовый к установке на жесткий диск. Это – 32-битная версия дистрибутива, хотя она пойдет без всяких проблем и на машинах класса x86-64, а при условии, что вашему компьютеру не больше пяти лет, Fedora 9 будет работать без запинок. Официально, минимальные требования таковы:

» Процессор 400-МГц Pentium II
» 92 МБ ОЗУ

Однако мы не советуем запускать Fedora на такой медленной машине – по крайней мере, ее надо приспособить (например, установить *IceWM* вместо Gnome). Для работы в Интернете, с офисом, графикой и для программирования рекомендуем 1-ГГц процессор и 512 МБ ОЗУ как минимальную исходную точку. Если вы все-таки решились установить Fedora 9 на старой машине, то всегда можете пойти по пути текстового режима. Так вы настроите компьютер для программирования или работы в качестве сервера, но тогда вам не будет доступен графический интерфейс. Требования для текстового режима таковы:

» 200-МГц Pentium CPU
» 128 МБ ОЗУ

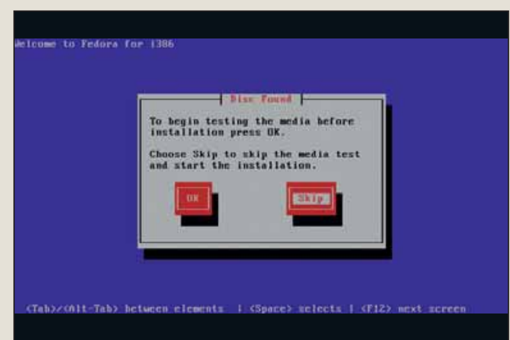


Шаг за шагом: Устанавливаем Fedora 9



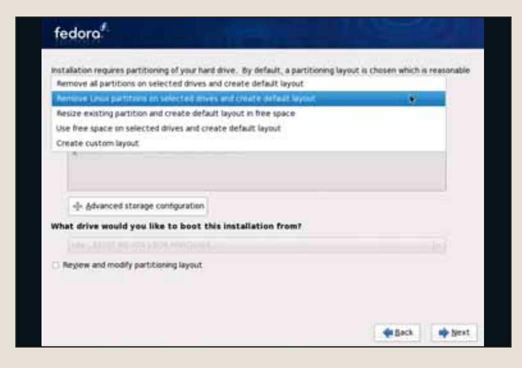
1 Загрузка

Загрузите ПК с **LXFDVD**, и, когда появится этот экран, нажмите Enter, чтобы начать установку, или стрелку вниз, чтобы выбрать текстовый режим.



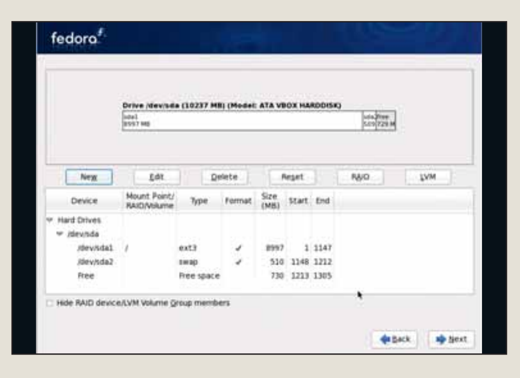
2 Проверка

Сейчас вы можете пропустить проверку оборудования, но если у вас возникнут проблемы, перезагрузитесь и проведите этот тест.



5 Деление на разделы

Задайте пароль root (администратора), затем выберите способ деления диска на разделы – см. текст выше, там вы найдете объяснение.



6 Вручную

Если вы выбрали пункт Custom, создайте корневой раздел (/) в формате ext3 размером не менее 5 ГБ, и раздел подкачки [swap] размером 512 МБ.

Каким бы путем вы ни пошли, вам понадобится не менее 5 ГБ свободного места на диске для стандартной установки, и 10 ГБ, если вы решили инсталлировать абсолютно все пакеты. Fedora загружается прямо с LXFDVD: на большинстве ПК надо только поместить диск в привод, перезагрузить машину и приступить. Однако если ваш ПК настроен на запуск с жесткого диска, а не с DVD, придется менять порядок загрузки в настройках BIOS.

Азбука разбивки диска

Во время установки вам предложат несколько вариантов разбиения диска – т.е. выделения места для дистрибутива. Если вы планируете отвести под Fedora всю машину, выберите Remove all partitions on selected drives and create default layout [Удалить все разделы с выделенных дисков и создать разбивку по умолчанию]. Вся информация будет утеряна, так что сперва создайте резервные копии важных данных! Если у вас двойная загрузка Windows и Linux и вы хотите заменить ваш текущий дистрибутив, подойдет Remove Linux partitions on selected drives and create default layout [Удалить разделы Linux с выделенных дисков и создать разбивку по умолчанию]. Старая установка Linux будет стерта, а Windows останется нетронутой – и после инсталляции вы по-прежнему будете иметь возможность загружать обе ОС.

Если вы – новичок в Linux, и в данный момент весь диск у вас отведен под Windows, выберите Resize existing partition and create default layout in free space [Изменить размер существующего раздела и создать разбивку по умолчанию]. После этого вы можете сжать часть диска, отведенную под Windows, высвободив 5 ГБ или больше для Linux. Как и при любых серьезных действиях с диском, следует прежде всего создать резервные копии важных файлов в Windows. И, наконец, если вы – линуксоид со стажем, и у вас уже имеется некоторый опыт (или вы действуете по принципу «на авось») и вам бы хотелось полностью управлять процессом разбивки диска, вы можете выбрать опцию Create custom layout [Создать разделы вручную] и сделать все самостоятельно.

Процесс установки после разбивки диска не должен вас озадачить – вам предложат выбрать раскладку клавиатуры, создать пользовательскую учетную запись и произвести прочие действия, присущие инсталляции дистрибутива. Но если что-то пойдет не так, у вас есть несколько вариантов:

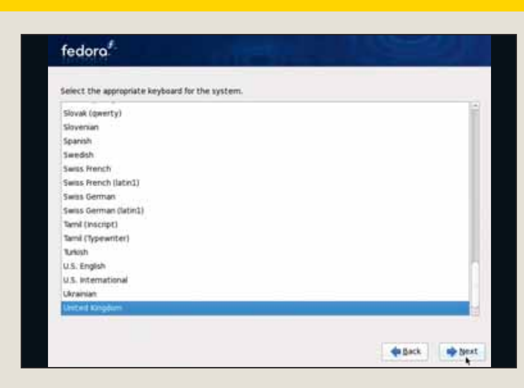
1 Перезагрузитесь и попробуйте снова. Выполните сканирование диска, как предлагает Fedora, чтобы проверить, в порядке ли ваш DVD.

2 Удалите ненужные периферийные устройства. Ясно, что без клавиатуры и мыши не обойтись, но если Fedora подвиснет, пытаясь распознать ваш принтер или сканер, отключите их на время – вы сможете добавить их позже.

3 Получите подсказку онлайн. Дружественные форумы на www.linuxforum.ru – то, что нужно для начала; разместите сообщение, в котором указывается, что именно не работает, и другие пользователи Fedora смогут помочь вам советами. Если это не поможет, зайдите на www.fedoraforum.org – там вы найдете множество подсказок по Fedora. Оставляя сообщения на форумах, постарайтесь привести всю необходимую информацию (например, спецификации вашего ПК, что вы пытались сделать, какое сообщение об ошибке вы получили, и т.д.) Чем больше информации, тем лучше! Но скорее всего, никаких проблем при установке и запуске Fedora у вас не возникнет, так что приятного вам знакомства с дистрибутивом и его новыми программами. »

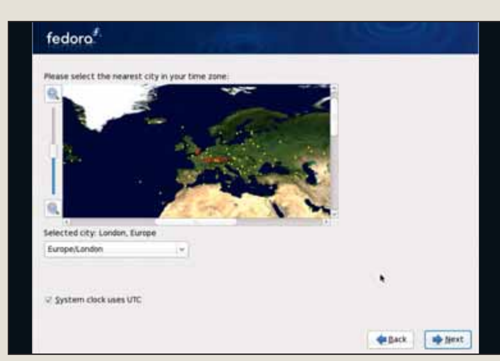


› **Установив Fedora, вы получите предложение создать учетную запись пользователя и отправить проекту отчет об оборудовании (анонимно и исключительно по желанию).**



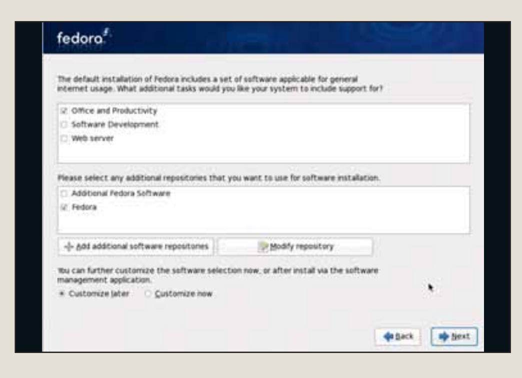
3 Раскладка

При появлении графической программы установки, нажмите Next, чтобы выбрать язык и раскладку клавиатуры.



4 Куда ставить?

Далее вы сможете указать, делать ли свежую установку или обновить существующую инсталляцию Fedora. После этого выберите сетевые настройки и часовой пояс.



7 Софт

После установки загрузчика вы можете изменить набор приложений. Если сомневаетесь, просто выберите набор по умолчанию – добавить новые приложения всегда успеется.



8 Ждите

Fedora начнет копирование на ваш жесткий диск (это может занять 30–60 минут). По окончании этого процесса ваш компьютер перезагрузится, так что извлекайте DVD и следуйте инструкциям по окончательной настройке.

Работа с Fedora 9

Завершив установку, можете войти в систему под созданным вами именем пользователя и паролем – вы попадете на рабочий стол Gnome. Если вы – ветеран Linux, то легко во всем разберетесь, а если нет – вот небольшой тур специально для вас. Панель меню вверху позволяет запускать программы, осуществлять навигацию по различным частям файловой системы (например, по домашней директории в поиске персональных данных) и настраивать дистрибутив. Вы можете перетаскивать файлы на рабочий стол, а внизу находится панель задач: здесь перечисляются запущенные программы. Справа от нее – переключатель рабочих столов: фактически он увеличивает площадь вашего экрана ровно в четыре раза, и можно, например, открыть интернет-приложения на одном рабочем столе, офисные приложения на другом, и т.д.

Fedora 9 включает *Firefox* в качестве web-браузера, *OpenOffice.org* как офисный пакет, *Pidgin* для обмена мгновенными сообщениями и *GIMP* для редактирования изображений. Загляните в меню Applications, и вы увидите множество других программ и утилит (да, и еще несколько игр, чтобы скоротать время в обеденный перерыв). Захочется добавить другие программы из обширных онлайн-репозиториев пакетов Fedora – откройте System > Administration > Add/Remove Software, и увидите список программ, объединенных по категориям. Выберите нужную категорию слева, щелкните по интересующей вас программе в списке справа, а затем нажмите на кнопку Install.

Вас также будут уведомлять о появлении обновлений. Fedora – дистрибутив, который не стоит на месте, так что исправления ошибок и заплатки безопасности появляются почти ежедневно. Дистрибутив просмотрит онлайн-репозитории на предмет обновлений; при их наличии замигает красная звездочка на верхней панели – нажмите на нее, и увидите, что нового появилось. Помните: обновления не обязательны, но, чтобы обезопасить вашу систему в режиме онлайн, рекомендуем по возможности все-таки их устанавливать.

Меню System содержит различные утилиты настройки для Fedora. Если вам надо сконфигурировать сетевую карту, изменить разрешение экрана, настроить принтер или добавить новых пользователей, зайдите в подменю Administration. А через Preferences можно настроить косметическую часть Fedora – в частности, загляните в Look and Feel > Desktop Effects, чтобы на вашем рабочем столе появились тени и 3D-эффекты, предоставляемые *Compiz*.



Перед тем, как вставить DVD в дисковод, пожалуйста, убедитесь, что вы прочитали, поняли и согласились с нижеследующим:

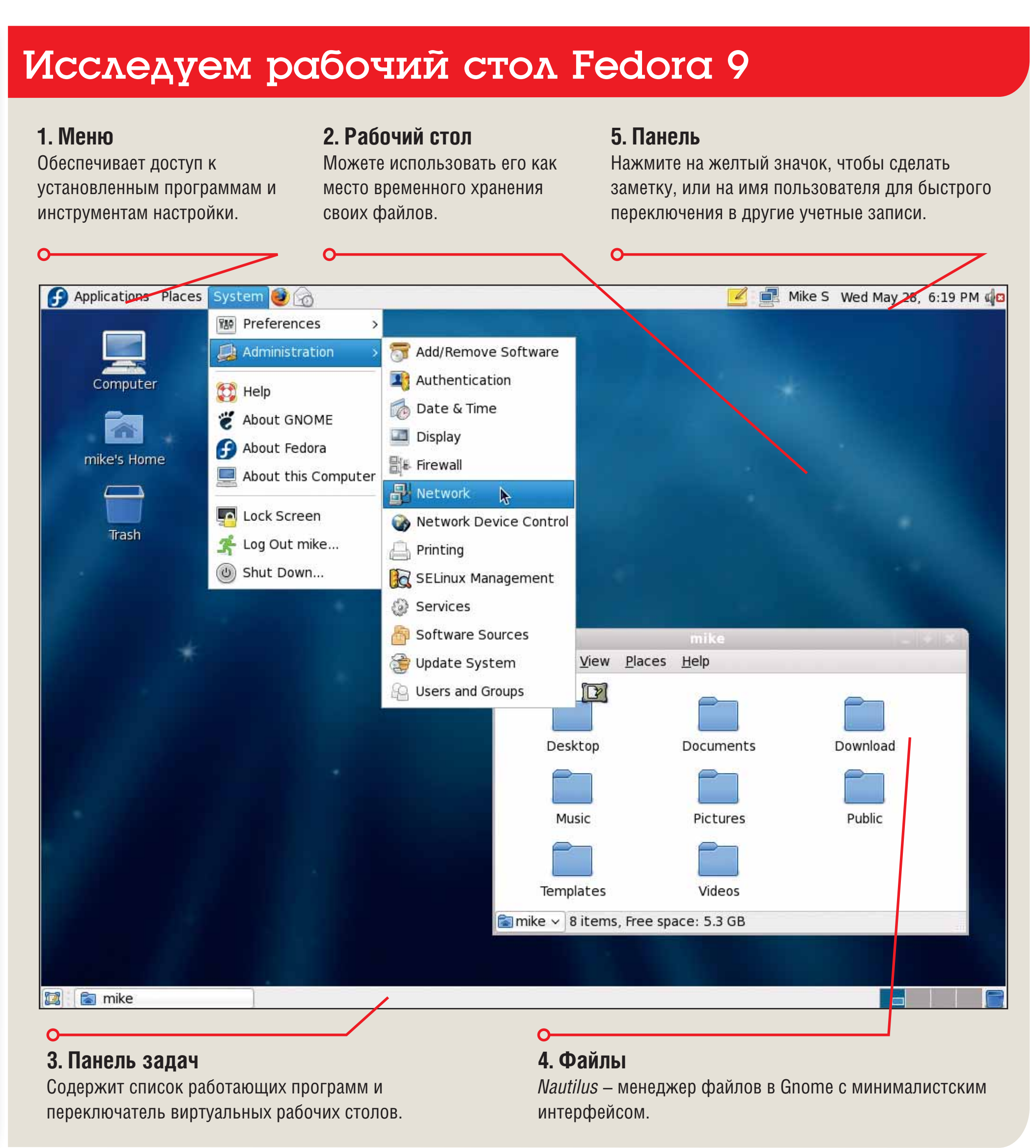
Диски *Linux Format* DVD тщательно проверяются на предмет отсутствия на них всех известных вирусов. Тем не менее, мы рекомендуем вам всегда проверять любые новые программы надежным и современным антивирусом.

Хотя процесс отбора, тестирования и установки программ на DVD проводится со всем тщанием, редакция *Linux Format* не несет никакой ответственности за повреждение и/или утрату данных или системы, могущее произойти при использовании данного диска, программ или данных на нем. Настоятельно рекомендуем вам создавать своевременные и надежные резервные копии всех важных файлов.

Чтобы узнать об условиях использования, просим вас прочесть лицензии.

Бракованные диски

В маловероятном случае обнаружения бракованного диска *Linux Format*, просим связаться с нашей группой поддержки по адресу disks@linuxformat.ru для получения содействия.



Исследуем рабочий стол Fedora 9

1. Меню
Обеспечивает доступ к установленным программам и инструментам настройки.

2. Рабочий стол
Можете использовать его как место временного хранения своих файлов.

5. Панель
Нажмите на желтый значок, чтобы сделать заметку, или на имя пользователя для быстрого переключения в другие учетные записи.

3. Панель задач
Содержит список работающих программ и переключатель виртуальных рабочих столов.

4. Файлы
Nautilus – менеджер файлов в Gnome с минималистским интерфейсом.

ВНИМАТЕЛЬНО ПРОЧТИТЕ ЭТО ПЕРЕД ТЕМ, КАК ИСПОЛЬЗОВАТЬ DVD-ДИСК !

Информация о диске

ЧТО-ТО ПОТЕРЯЛИ?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials» (Главное) на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, следует заглянуть именно туда.

ФОРМАТЫ ПАКЕТОВ

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любые другие. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными бинарными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы смогли собрать его самостоятельно.

ДОКУМЕНТАЦИЯ

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

ЧТО ЭТО ЗА ФАЙЛЫ?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux, различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

имя_программы-1.0.1.i386.rpm – вероятно, это бинарный пакет RPM, предназначенный для работы на системах x86;

имя_программы-1.0.1.i386.deb – такой же пакет, но уже для Debian;

имя_программы-1.0.1.tar.gz – обычно это исходный код;

имя_программы-1.0.1.tgz – тот же файл, что и выше по списку: «tgz» - это сокращение от «tar.gz»;

имя_программы-1.0.1.tar.bz2 – тот же файл, но сжатый bzip2 вместо обычного gzip;

имя_программы-1.0.1.src.rpm – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;

имя_программы-1.0.1.i386.FC4.RPM – бинарный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;

имя_программы-1.0.1.ppc.Suse9.rpm – бинарный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;

имя_программы-devel-1.0.1.i386.rpm – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru





Создание установочных дисков при помощи CDRECORD

Самый быстрый способ записать ISO-образ на чистую матрицу – это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Для начала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке, например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле **/etc/default/cdrecord**. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке – это метка, затем, после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь вы можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root, выберите вкладку Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на Combust!. Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecord* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере. На нем может стоять Windows, Mac OS X, AmigaOS, или любая другая ОС.

Нет устройства для записи дисков?

А что, если у вас нет устройства, с помощью которого можно было записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не обязательно использовать Linux для записи дисков, подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт его разработчика. LXF



Содержание LINUX FORMAT DVD Август 2008

LINUX FORMAT

Сторона 1

Поставляется вместе с журналом LINUXFORMAT номер 8 (108) Август 2008

Сторона 2

Комментарии? Присылайте ваши мысли и предложения по электронной почте: info@linuxformat.ru
Пожалуйста, ознакомьтесь с опубликованной в журнале инструкцией перед использованием данного диска.
Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать антивирусный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не может принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые может повлечь за собой использование этого DVD, представленных на нем программ или данных. Перед тем, как устанавливать какое-либо ПО на компьютер, подключенный к сети, проконсультируйтесь с сетевым администратором.

Дефектные диски. В маловероятном случае обнаружения дефектов на данном диске, пожалуйста, обращайтесь по адресу: disks@linuxformat.ru

Тираж изготовлен ООО «Фортмедиа», Россия, Санкт-Петербург, 196006 ул. Цветочная д. 7, тел. +7 (812) 388-8290. Лицензия МПТР России ВАФ № 77-225

Новое ПО

Рабочий стол и система

PeaZip – один из лучших инструментов архивирования, и мы с нетерпением ждали нового релиза 2.1; вы найдете его в разделе **Рабочий стол** нашего DVD. Поддерживая аж 32 формата сжатия, *PeaZip* способен открыть или создать практически любой архив, который можно вообразить, и с ним даже можно работать в Windows, когда Linux недоступен. После выхода релиза 2.0 разработчики потрудились над тем, чтобы сделать *PeaZip* утилитой управления файлами более общего назначения – теперь вы можете перемещать, копировать и удалять файлы независимо от того, находятся ли они внутри файловой системы или в сжатом архиве.

Весьма удобно, что команда *PeaZip* предусмотрела для последней версии deb- и RPM-пакеты, так что вам не придется компилировать программу из исходных текстов. Скопируйте нужный пакет в домашнюю директорию и откройте окно командной строки. Для пакетов .deb (например, в системах Debian или Ubuntu), переключитесь на root и введите:

```
dpkg -i peazip*.deb
```

Если вы используете систему на RPM, например, Fedora, Mandriva или OpenSUSE, введите

```
rpm -ivh peazip*.rpm
```

Раздел **Help/Новичку_в_Linux** на DVD содержит более подробную информацию по установке пакетов.

В *PeaZip* вы можете осуществлять навигацию по файловой системе; при нажатом Shift выделяются несколько файлов. Затем нажмите Add на панели инструментов, чтобы создать новый архив, указав его формат во вкладке Options. Для переключения с отображения файловой системы на вывод файлов, которые надо заархивировать, и наоборот, нажмите на зеленую стрелку вверху справа в основном окне. А пока вы в разделе **Рабочий стол**, попробуйте также *Recoll*, систему «персонального поиска по тексту» для KDE. Эта программа сканирует множество различных форматов документов, от простого текста и HTML до *OpenOffice.org* и *Scribus*, что поможет вам быстро и легко находить фрагменты информации. Она немного похожа на *Beagle* и Google Desktop Search, но концентрируется на текстовом содержимом. К счастью, ей не нужен фоновый демон базы данных, сжирающий память и процессорное время. Учтите, разработчики рекомендуют компилировать *Recoll* с *Qt 4.3* или более ранней версии, поскольку 4.4 вызывает довольно много ошибок.

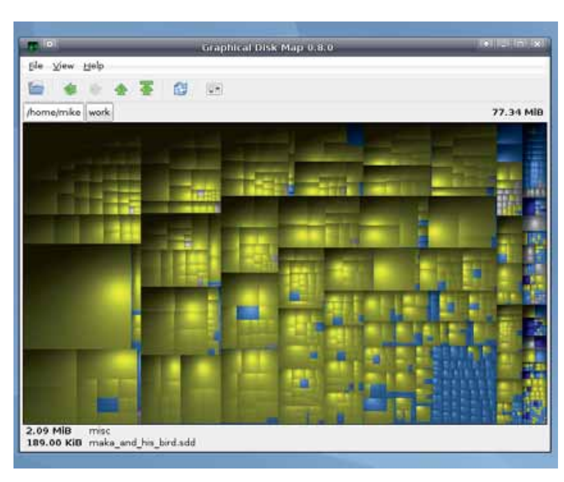
Карты на стол

Раздел **Система** содержит *GdMap* или Graphical Disk Map – утилиту, которая создает визуальное представление вашего жесткого диска со всем его содержимым, чтобы вы с одного взгляда могли сказать, какие файлы занимают больше всего места. На данный момент она доступна только в виде исходного кода, так что скопируйте файл **gdmap-0.8.1.tar.gz** в домашнюю директорию и распакуйте его. Перейдите в полученный каталог и введите:

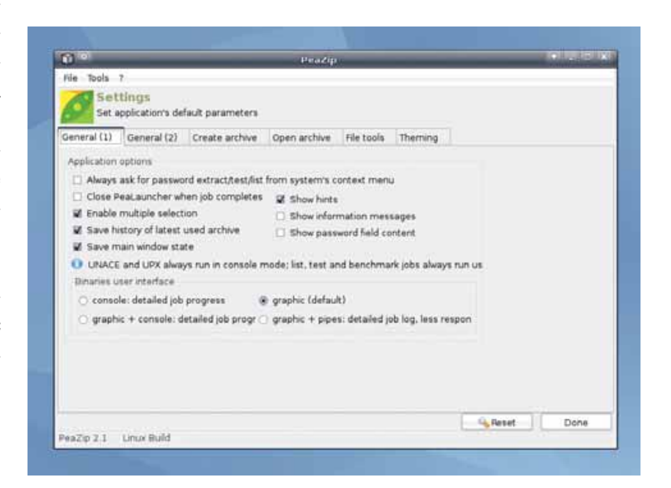
```
./configure
make
make install
```

(Третью команду надо набирать от имени root; если вам это ничего не говорит, загляните в раздел **Help/Новичку_в_Linux** на DVD.) Затем введите **gdmap**, чтобы запустить *GdMap*. Вы увидите основное окно черного цвета – нажмите значок Open a folder [Открыть папку] на панели инструментов, чтобы выбрать директорию для сканирования. В зависимости от ее размера, этот процесс займет от нескольких секунд до нескольких минут, а когда он закончится, вы увидите результат, подобный приведенному на экранном снимке внизу.

Каждый квадрат обозначает файл: чем он крупнее, тем больше файл. Можно навести на файл мышь, чтобы узнать его точный размер, и дважды щелкнуть по нему, чтобы узнать, в какой директории он находится. Для возврата в родительскую директорию нажмите кнопку Back на панели инструментов. *GdMap* – отличный маленький инструмент для поиска в домашней директории позабытых там слишком больших файлов, без пачканья рук в командной строке.



› Отстреливайте огромные и бесполезные файлы на жестком диске – охота ведется мощным оружием *GdMap*.



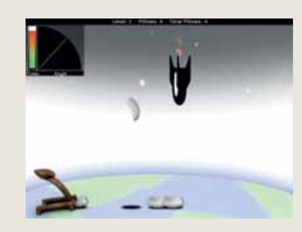
› *PeaZip* – самый мощный и настраиваемый архиватор для Linux.

Дополнительно

Игрушки

Наш раздел **Игры** открывает *Too Hard For You*, обычно скрывающаяся за аббревиатурой 2H4U. Это – причудливая смесь элементов *Tetris* и *Breakout*, игровой процесс которой призван напрячь ваши способности к многозадачности. Используйте клавиши W, A, S и D для управления блоками в стиле *Tetris*, которые должны образовать сплошные линии, а мяч управляется клавишами со стрелками. Сначала это может показаться очень странным, и вам, возможно, захочется сбежать, взвыв от сложности – но, поиграв немного, вы почувствуете, что уже находитесь «в теме».

Кроме того, у нас есть *Pillows* [*Подушки*] – обманчиво простая, но затягивающая игра, которую нельзя отнести ни к одному из существующих жанров. Сюжет таков: неизвестный агрессор подверг Землю ракетному удару. Ваша задача – спасти мир от ужасной гибели, используя гигантскую катапульту и «до смешного пухлые» подушки, чтобы заслонить мир от этих ракет. Все, что вам нужно – клавиша пробела: смотрите на индикатор сверху слева, жмите на пробел, выбрав нужный угол, и удерживайте клавишу, пока не достигнете нужного уровня мощности. Вы стреляете подушкой, и если она приземлилась на том месте, где должна была упасть ракета, мир спасен (а вы переходите на следующий уровень). LXF



› Мягкий и приятный способ спасения мира – *Подушки* [*Pillows*].

Лучшие новинки открытого ПО на планете

LXF HotPicks



Ричард Смедли
Активист сообщества FOSS и давний сотрудник **LXF**, Ричард бороздит глубины с умницей-дельфином в поисках сокровищ **HotPicks**.

В ЭТОТ РАЗ ТОЛЬКО ДЛЯ ВАС: Liferea » Qtractor » Strasheela » Endeavour II » Wordwarvi » Cave9 » ColorExplorer » TuxType » When DelSafe

Новостной агрегатор

Liferea

LINUX FORMAT Hottest Pick

Версия 1.4.15 **Сайт** http://liferea.sourceforge.net

Liferea (Linux Feed Reader) – идеальное решение для сбора новостей из блогов, подкастов и сайтов, на просмотр которых у вас не хватает времени.

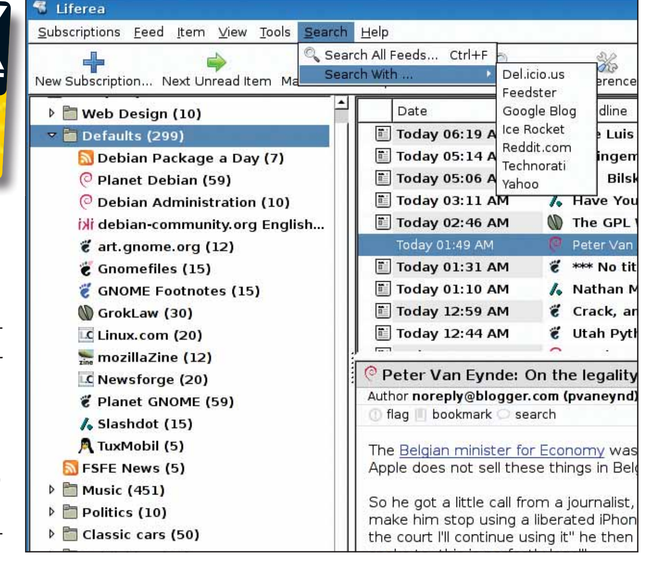
Ленты RSS и Atom – самый распространенный в Web способ привлечения вашего внимания к обновлениям, будь это новостное сообщение, комментарий или что угодно еще – естественно, при условии, что у вас есть программа-агрегатор. *Liferea* выполняет эту работу быстро и без всяких проволочек.

Liferea – приложение *GTK*, и оно отлично вписывается в среду Gnome; однако это не повод не применять его на рабочем столе KDE в качестве альтернативы *Akregator* – особенно если вы используете *Firefox*, а не *Konqueror* (*см. ниже*). У него всего несколько зависимостей, и его легко установить, хотя ваш менеджер пакетов, вероятно, и так сделает за вас всю работу, как Ubuntu сделал за нас.

Просматривайте

Liferea показывает новости с помощью встроенного браузера (*см. экранный снимок справа*), но его можно настроить на запуск любого обозревателя по вашему выбору. Интеграция с браузером работает в обе стороны, и *Firefox* подпишет вас на новостные ленты, если вы выберете Feed Subscription в меню настроек. В *Epiphany*, браузере по умолчанию среды Gnome, связывание происходит автоматически.

Интерфейс организован просто, через меню или опциональную панель инструментов. New Subscription позволяет ввести полный URL новостной ленты, а можно просто указать URL сайта *Liferea* и разрешить ему искать ленты новостей для вас. Новости просматриваются через трехпанельный интерфейс, снискавший повсеместную любовь почтовых клиентов.



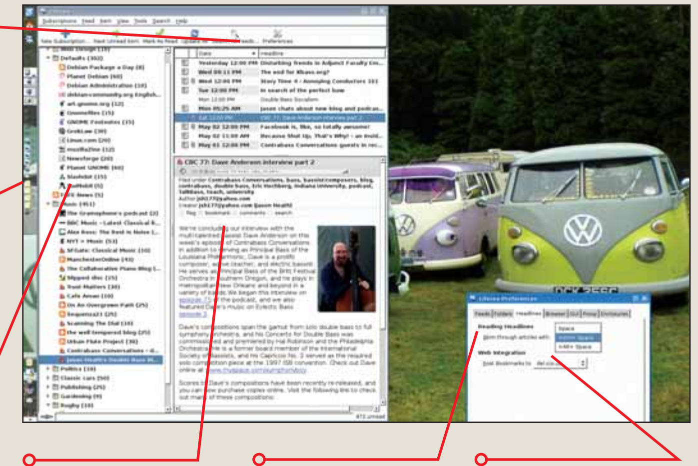
› Можно валить все новости в одну папку и смотреть их разом, но это не задано по умолчанию: одновременная загрузка множества лент очень медленная.

Нажатие на родительскую папку в левой панели откроет список подпапок или подписок на новостные ленты, а в окне Preferences вы сможете выбрать просмотр всех лент в объединенной папке. Это создает известную неразбериху, но ее можно уменьшить, приказав скрывать уже прочитанные сообщения. Другие опции контролируют число сохраняемых новостей и частоту их обновления в *Liferea*, и вы даже можете выбирать, какой социальный сайт следует использовать, если вы решите добавить ссылку на новость в закладки правым щелчком мыши.

Виртуальные папки – подобные таковым в *Evolution* – создаются из поисковых запросов, что позволяет просматривать постоянно запрашиваемую информацию так же легко, как и любую другую ленту. Это очень полезно при отслеживании вашего любимого проекта, или интересной подборки новостей, или горячей темы в сообществе свободного ПО.

Со времени наших последних обзоров в LXF60 и LXF87 *Liferea* сделала гигантский скачок вперед. Есть множество других агрегаторов, но ни один из них не обладает такой простотой, скоростью и продвинутыми возможностями фильтрации.

Исследуем интерфейс Liferea



Поиск
Найдите нужную новостную ленту, запустив поиск по нескольким популярным сайтам; можно также искать в сохраненных новостях.

Значки лент
Многие сайты используют пиктограммы, упрощающие выбор новостей. К сожалению, есть и другие, использующие настройки по умолчанию таких провайдеров, как Bloglines.

Подпапки
Иерархия папок в *Liferea* – отличная возможность организовать между собой разные сферы интересов.

Обозреватель
Просматривайте новости в одном из встроенных браузеров или позвольте *Liferea* открыть ваш любимый внешний web-обозреватель.

Настройка
Liferea предоставляет вам ровно столько настроек, сколько нужно, чтобы она работала именно так, как вам хочется.

Закладки
Похоже, что web-интеграция позволяет выбрать любой социальный сайт с закладками, какой только существует под солнцем.



Аудио/MIDI-секвенсор

Qtractor

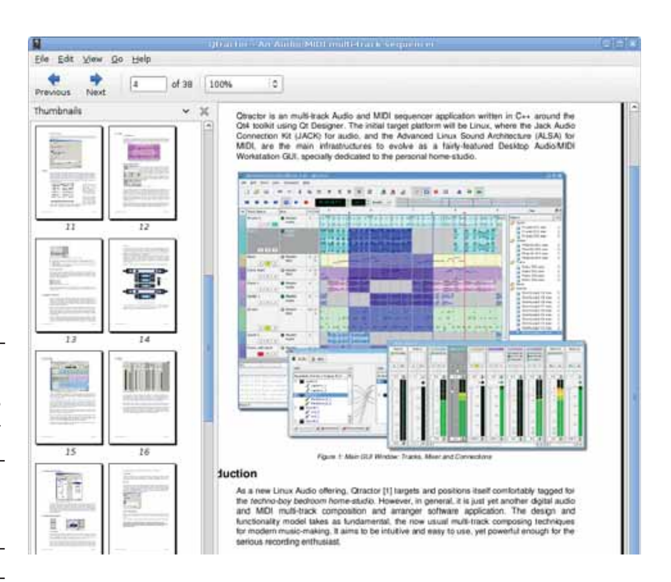
Версия 0.1.3 **Сайт** http://qtractor.sourceforge.net

Мощность домашнего ПК открывает доступ к такой работе со звуком, о которой The Beatles могли только мечтать. *QTractor* – мультитрековый «комнатный секвенсор», уместный также и в профи- и полупрофи-установках, благодаря своему доходчивому интерфейсу, где использованы принятый в индустрии стандарт мультитрекового просмотра композиций и простая интеграция в аудиосреду ALSA и JACK. Каждые несколько лет звукозапись в Linux делает эволюционные рывки – и, похоже, мы наблюдаем очередное изменение; но все основные дистрибутивы сейчас поставляют ALSA, а JACK – самый популярный и эффективный способ «собрать» виртуальную студию, так что для установки и запуска *QTractor* не потребуется ничего экзотического. Хотя, возможно, вам придется покопаться в репозиториях вашего дистрибутива в поисках опциональных пакетов, например, librubberband, позволяющего увеличить время звучания и изменить тональность (разработка Криса Кэннема [Chris Cannam], снискавшего славу благодаря *Rosegarden* и многим другим аудио пакетам – см. www.all-day-breakfast.com/cannam).

Подобная обработка звука – это лишь верхушка весьма солидного айсберга, но вот редактирование MIDI и аудио, а также способности секвенсора могут превратить его в центр вашей станции по обработке цифрового звука (DAW). Аудиотреки можно импортировать из большинства популярных форматов, а инструменты записываются через JACK. Редактирование осуществляется простым перетаскиванием или с помощью настраиваемых горячих клавиш, и даже модули расширения можно легко копировать на разные треки.

Феноменально продуктивный Руи Нуно Капела [Rui Nuno Capela] – он же автор *QSynth*, *QSampler* и *QJackCtl* – заявляет о нехватке времени для работы над *QTractor*. Однако же последний релиз, 0.1.3, под кодовым названием «frugal damsel» («экономная барышня»), демонстрирует значительные улучшения по сравнению с состоянием на момент выхода номера **LXF101**), в частности, в редактировании MIDI и поддержке VST-модулей. В нем также имеется (опциональная) поддержка *Rubber Band Audio Time Stretcher Library*.

QTractor снабжен 38-страничным руководством пользователя; если вы когда-либо использовали DAW-приложения, вряд ли что-то будет представлять для вас трудность, а вот новичкам его следует почитать!



› Руководство: получше многих, поставляемых с коммерческими приложениями. Секвенсор: готов к испытаниям в вашей DAW.

Композитор

Strasheela

Версия 0.9.5 **Сайт** http://strasheela.sourceforge.net

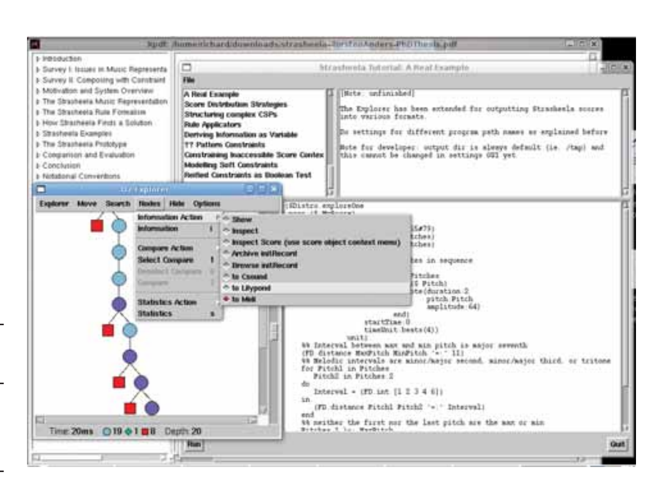
Компьютерные модели композиции не являются новинкой, но *Strasheela* предлагает подход, который, хотя и требует много времени на обучение, весьма эффективен в большинстве формальных систем. Не-музыкант и не-программист будет биться со *Strasheela*, однако спец в музыке, хоть и не-программист, вскоре откроет для себя возможности подобного общего подхода к CSP (Constraint Satisfaction Problems, задачи поиска допустимого решения – интерфейс между установленными правилами композиции и неизвестными переменными), и благодарить за это надо всеобъемлющую документацию.

Strasheela основан на системе Mozart и потому написан на мощном объектно-ориентированном языке Oz; вы найдете руководство для начинающих в **~/.oz/<version>/bin**. Учебник открывается в *Emacs*, с которым *Strasheela* очень тесно связан. Пусть поклонники Vi и расстроятся, но это весьма разумно, когда дело доходит до гармонизации приложения со скриптовым языком, набором документов и различными выходными фильтрами для всяческих переносимых форматов.

Strasheela позволяет перепрограммировать любое ограничивающее условие (rule program) путем полного отделения определения правила от его применения, что является параллелью способности *Emacs* переписывать любой раздел редактора на лету, унаследованной с самого его рождения на машинах Lisp в MIT в погоне за искусственным интеллектом. Если это намного превышает ваши ожидания, касающиеся лишь элементарных музыкальных экспериментов, не пугайтесь: есть и простой (относительно) путь к основанной на математике музыке *Strasheela*.

Запустите руководство и познакомьтесь с основными примерами, чтобы разобраться в синтаксисе, затем перейдите к более крупным примерам с сайта, вроде генератора контрапункта реального времени. Попробуйте изменить несколько переменных – вы не превратитесь в Баха, но все же в результате, возможно, появятся несколько приятных для слуха музыкальных фрагментов, которые можно будет оставить как есть или взять за основу для более амбициозного проекта.

Экспорт в Csound, LilyPond или MIDI весьма упрощает этот шаг, и вы можете записать результаты своих экспериментов посредством вывода в PostScript или PDF. Прочие функции доступны через различные дополнительные приложения и некоторые расширения Oz – все это описано в файле **Installation**. Возможно, это слишком узкоспециализированное приложение, но человеку, причастному к музыке, оно может открыть новые миры.



› Дайте волю своему Моцарту — вам поможет эта причудливая комбинация музыки и программирования.

Файловый менеджер

Endeavour II

Версия 2.9.2 **Сайт** www.battlefieldlinux.com/wolfpack/Endeavour2

Файловый менеджер нередко служит главным связующим звеном между вами и вашим компьютером. Как правило, вы западаете на менеджер, имеющийся в вашей ОС – обычно это *Konqueror* или *Nautilus* – и нужны весьма серьезные причины, чтобы вы его сменили. Что ж, мы, возможно, нашли их для *Endeavour II*.

Те, кто обходит стороной объемистые KDE и Gnome, предпочитают файловые менеджеры полегче – и остаются без дополнительные удобств: например, просмотра изображений или архивирования. *Endeavour II* предоставляет в ваше распоряжение все эти инструменты плюс утилиты для устройств и дисков, и скоростную программу, согласную работать на более скромных системах, чем требуются KDE/*Konqueror* и Gnome/*Nautilus*.

Помимо двухпанельного браузера в стиле дерево-и-список вы получите полезное средство для просмотра изображений, гибкий архиватор, скачивание методом Drag and Drop (через *Wget*), поддержку MIME, а также разнообразные инструменты и утилиты.

Две панели – левая, отображающая директорию, и правая, выводящая ее содержимое – будут знакомы пользователям Проводника Windows, собственно, как и весь графический интерфейс, включая меню. А еще одна общая черта с Windows – это корзина: все, что удаляется в *Endeavour*, оказывается здесь, допуская восстановление.

На сайте имеются пакеты для многих дистрибутивов, а также расширения, позволяющие сделать все, что захотите – от управления приводами Iomega Zip и Jazz до форматирования дискет.

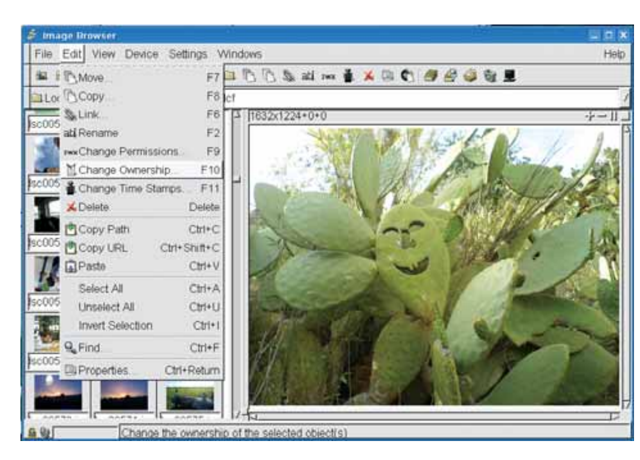
Если разговоры о старых носителях попахивают для вас ретроградством, приготовьтесь к еще большему шоку: *Endeavour II* использует библиотеки *GTK 1.2*. Коль скоро ваше сердце отдано современным инструментам GUI с закругленными виджетами и тенями, это приложение сразу же лишится вашего внимания, но целевую аудиторию тех, кто не является пользователями KDE/Gnome, оно вряд ли оттолкнет. В любом случае, это дело вкуса.

Только посмотрите

Одна из лучших функций *Endeavour II* – программа просмотра изображений. Левая панель отображает миниатюры, а правая – полноразмерный вариант выделенной картинки. Изображения можно изменять в размерах и поворачивать, как обычно.

В установке по умолчанию отсутствует пакетная обработка графики, но вы можете скачать нужную программу непосредственно со страницы проекта «Contrib». Доступны также дополнительные значки, оболочка для *ClamAV* и подборка MIME-типов – все это восполняет недостатки стандартной инсталляции.

> «Endeavour II намерен стать GUI-эквивалентом Midnight Commander».

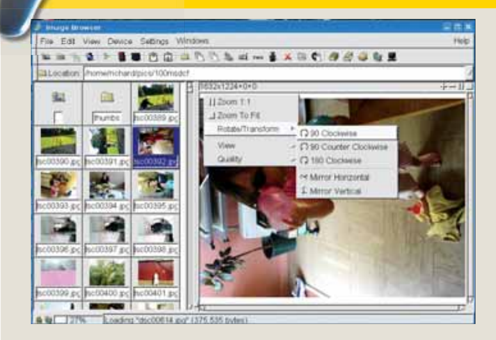


› *Endeavour II* даст вам GUI для команд типа chown.

Некоторые опции по умолчанию сперва кажутся странноватыми – например, выбран почтовый клиент *Balsa*, несмотря на тот факт, что весь мир и его матушка используют *Evolution*. Поскольку текстовый редактор по умолчанию – *Vi*, многие захотят немного повозиться с опциями, чтобы приспособить *Endeavour II* к повседневным нуждам. Сделав это, вы найдете отзывчивый менеджер файлов с набором дополнительных функций, воспринимаемых пользователями *Nautilus* и *Konqueror* как нечто само собой разумеющееся, но отсутствующих в более легковесных альтернативах.

Endeavour II претендует на роль GUI-эквивалента текстового менеджера файлов *Midnight Commander* – быстрого, всестороннего и простого в использовании. Людям, у которых много работы, он отлично подойдет – проведите с ним день и посмотрите, являетесь ли вы одним из них.

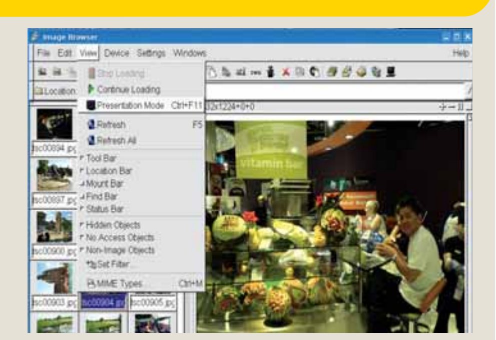
Шаг за шагом: Endeavour II



» Оглядитесь
Endeavour II снабжен полноценным просмотрщиком изображений, предлагающим обычный набор преобразований для перевернутых изображений.



» Фильтрация по типу файла
Вы можете находить изображения в большой директории, или отделить PNG от JPEG, если фотографии и экранные снимки хранятся у вас вперемешку.



» Режим презентации
Браузер для просмотра изображений имеет удобный режим презентации, на случай, если вам захочется составить слайд-шоу из своих изображений.

HotGames Развлекательные приложения

Стрелялка

Word War Vi

Версия 0.09 **Сайт** http://wordwarvi.sourceforge.net

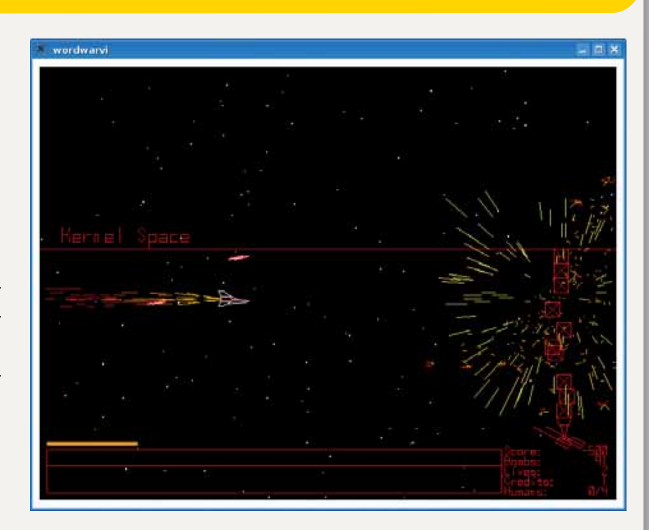
Аркады старой школы хороши и сами по себе, но уж если к ним приложит руку мастер, они становятся просто неотразимыми. *Word War Vi* – это стрелялка, позволяющая по-новому взглянуть на старый спор *Vi* с *Emacs*.

Вы должны провести свой корабль по основной памяти, чтобы спасти утраченные SWP-файлы. Во время своей миссии милосердия вы должны пробить защиту ОС и изгнать «процессы *Emacs*, пожирающие память». Отсюда вполне понятно, что автор занимает в войне редакторов четкую позицию. К счастью, у него есть также и чувство юмора, впрочем, как и у большинства любителей *Emacs* в штабе **LXF** – должно быть, оно появилось у нас в добрый час.

Вы начинаете игру, нажав на Q (видимо, потому, что с этой буквы начинается слово «quarter» – «четвертак», т.е. 25 центов, монета, необходимая, чтобы начать настоящую аркаду). Вообще-то управление с клавиатуры весьма своеобразно – в нем использованы традиционные для *Vi* кнопки HJKL (взамен можно использовать стрелки курсора), Z или пробел, чтобы послать сигнал -9 (лазер), B – чтобы внедрить указатель NULL в исполняемый поток (сбросить бомбу), и C – чтобы ускользнуть от преследователей (забросав их мусором).

Чтобы установить игру, распакуйте архив с исходниками (tar xvf имя_пакета.tar.gz), перейдите в директорию с помощью *cd* и затем введите *./configure*, после чего – *make install* от имени root; если вы побаиваетесь, см. раздел **Новичку** в Linux на **LXFDVD**.

Это – очень старомодная аркада, с шумной стрельбой и взрывами. Удовольствия добавляет саундтрек в стиле рок, записанный самим автором и его друзьями. По мере продвижения от уровня к уровню, неистовство нарастает.



› Полет над пушками может привести вас опасно близко к пространству ядра.

Разработка ведется очень быстро, и исправления ошибок появляются в CVS еженедельно. Если вы извели свою юность на аркады или на 8-битные ПК и консоли, то вас не придется уговаривать приступить. Если же игры 80-х прошли мимо вас, возможно, пора с ними познакомиться – по крайней мере, будет что ответить, если вас вдруг спросят: «А что ты делал во время Войны Редакторов, папа?»

Игра-исследование пещеры

Cave9

Версия 0.2.2 **Сайт** http://code.google.com/p/cave9

Вообразите, что вы удираете на своем истребителе от врагов, и в последнем отчаянном порыве решаете нырнуть в огромную пещеру на склоне горы перед вами.

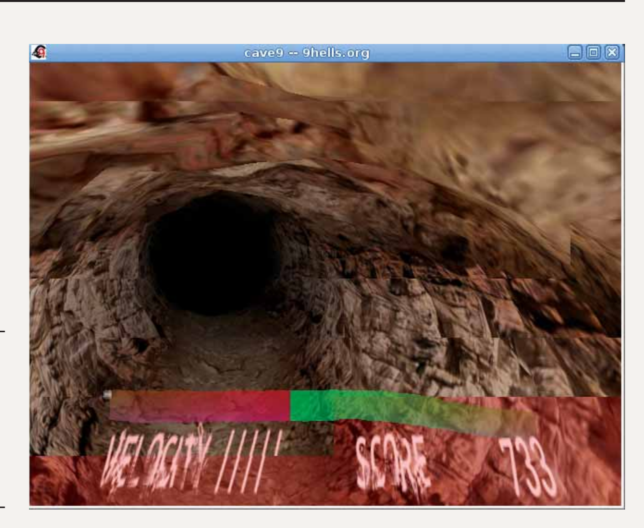
В миг, когда корабль входит в пещеру, вы обнаруживаете, что у него отказало управление, и скорость снизить нельзя. Лишь два двигателя реактивной тяги способны предотвратить столкновение со стенами пещеры и дают слабую надежду на то, что где-то есть выход и вы сможете его найти. От гибели и от кроваво-красной надписи GAME OVER вас спасут только кнопки со стрелками.

Если это звучит знакомо, значит, вы уже сталкивались с онлайновыми или мобильными версиями *SFCave*, которой отдает дань *Cave9*. Изначальная 2D-версия была игрой с управлением одной кнопкой, а основанная на Flash *3D-SFCave*, в которую можно играть онлайн, выглядит сыровато рядом с созданными Карло Капуто [Carlo Caputo] и Родольфо Борхесом [Rodolfo Borges] текстурированными пещерными проходами в *Cave9*.

Погружаемся

Управление очень простое. Стрелки влево и вправо контролируют диагональные реактивные двигатели, а стрелка вверх позволяет управлять ими обоими сразу, чтобы взлететь. Такого рода простые в управлении игры идеальны для мобильных устройств и для того, чтобы под рукой было средство для десятиминутного отдыха от работы.

Установка проста: используйте менеджер пакетов, чтобы добавить библиотеки SDL (версию для разработчика), скачайте исходники с Google Code, распакуйте и запустите make (от имени root). Написанная на C, с небольшим количеством связующего кода для работы с настройками на Perl,



› Наслаждайтесь простыми радостями отсрочки от гибели благодаря своим реактивным двигателям.

Cave9 – это дар сообщества тем, кому интересно учиться. Код достаточно легко читается и использует вездесущие библиотеки SDL, что превращает его в идеальный объект изучения для потенциальных разработчиков игр. А если вы не особо рветесь программировать на C, *Cave9* даст вам отличную возможность развлечься в ожидании электрички, почувствовав себя Хэном Соло или Люком Скайуокером.

Подбор цветов

Color Explorer

Версия 1.8 **Сайт** http://billposer.org/Software/ColorExplorer.html

Нашли отличный оттенок пурпурного и хотите его запомнить, чтобы использовать в своем сайте о завоевании Британии римлянами? Расширения браузера способны захватить образчик с web-страницы, но чтобы взять цвет с одного пикселя на рабочем столе, нужен *Color Explorer*.

Идеально подходящий для тех из нас, кто знает, чего хочет, но не всегда знает, как это получить, *Color Explorer* позволяет взять нужный цвет одним нажатием в любой точке рабочего стола, предоставляя палитру RGB (Red, Green, Blue) для последующего воспроизведения оттенка в любом графическом приложении или на web-странице.

И это еще не все. Используя бегунки, можно генерировать собственные смешения цветов, обучаясь по ходу действия. По умолчанию палитра RGB дана в шестнадцатеричной системе, но если вы предпочитаете десятичную, стоит лишь кликнуть. Можно работать и иначе – вставлять в приложение десятичные или шестнадцатеричные числа и смотреть, какой цвет получится в результате.

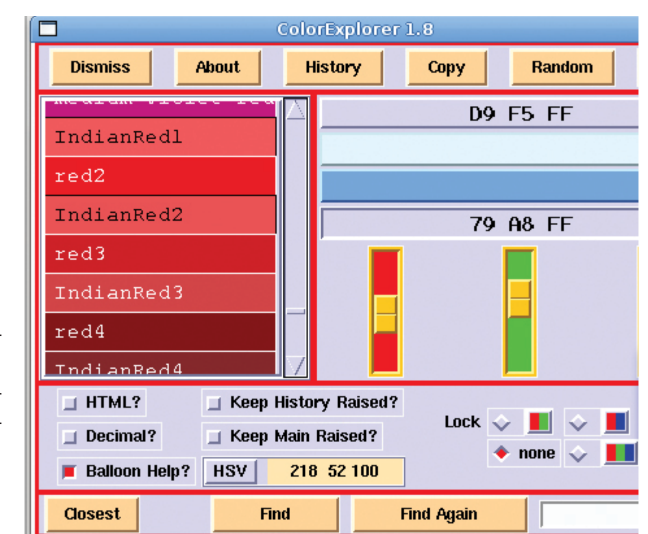
Если же вы предпочитаете цветовые диаграммы, прокручивайте виджет слева и выбирайте из огромного количества цветов. Можно даже осуществлять поиск по этому широкому диапазону, используя простое регулярное выражение, если вам ближе заумный подход. А если вам и этого мало, воспользуйтесь случайным выбором.

Сравнение и контраст

Палитра показывает два цвета сразу, и вы можете постепенно менять оттенок, сравнивая результат с первоначальным цветом. Есть очень интересная функция – можно связать два выделенных цвета, и затем изменять один, двигая бегунок. Другой цвет будет изменяться в той же мере, что и первый, давая оценку взаимосвязи в данной паре. И это занятие, и цветовой штурвал подарят вам несколько часов удовольствия, разрабатываете ли вы интерфейс или планируете цветовую гамму для своего офиса.

Установка прямолинейна благодаря минимуму зависимостей. *Color Explorer* требует наличия языка Tcl и набора инструментов Tk (и программы *Xoris* для копирования цветов с рабочего стола; все это имеется в исходных текстах и компилируется автоматически).

Помимо перехода на GPL v3, этот свежий релиз обзавелся улучшенной справкой и всплывающими подсказками по инструментам, сохранив при этом простоту интерфейса. Это – несложный инструмент, но очень многие найдут его весьма полезным для самых различных целей.



› Нужны цвета? *Color Explorer* запросто найдет их и поможет дать им названия.

Тренажер клавиатуры

TuxType

Версия 1.5.17 **Сайт** http://tux4kids.alioth.debian.org

Сейчас в школах редко учат набору текста, хотя в наш век это одно из ключевых умений (по крайней мере, пока мы не усовершенствовали способ ввода в стиле *Star Trek*). Многие ИТ-профессионалы по-прежнему печатают хоть и быстро, но двумя пальцами, и отнюдь не желают переучиваться на слепой метод ради увеличения скорости набора. Требуется мотивация: *TuxType* – это забавный способ освоения основных навыков. *TuxType* написан для более юных учеников, и взрослые, возможно, решат, что это не для них, но тут действительно добавлено веселья в то, что большинство считает нудным учебным процессом.

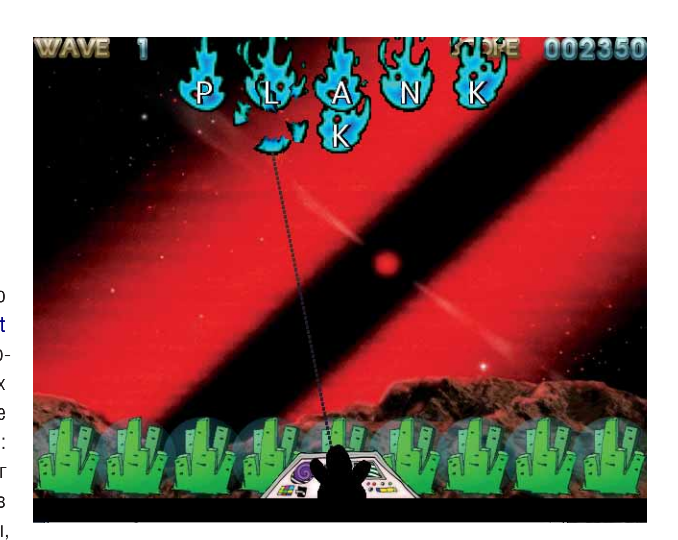
Режим практики из меню Options предлагает вам напечатать предложение, подсказывая, каким пальцем набирать, что очень удобно, как только вы сдвигаетесь с опорного ряда ASDFGHJKL. Меню Lessons [Уроки] содержит традиционные упражнения, такие, как lad lass dad fad had и т.д., чтобы вы привыкли печатать, не глядя на клавиатуру. Но переключитесь на Fish Cascade или Comet Zap – тут-то и начнется веселье. Мы протестировали *TuxType* на нескольких детях и взрослых и обнаружили, что самые юные ученики в восторге от игры Fish Cascade: надо печатать слово на рыбе, чтобы ее смог проглотить Тукс, который носится вокруг в ожидании этого момента. Доступны буквы, слова и упражнения, и вы можете добавлять в игру свои собственные слова и озвучку, следуя указаниям в документации.

Завоеватели пробела

«Comet Zap» демонстрирует, как Тукс бежит по падающим словам в то время, как вы печатаете их, и этот сценарий очень напоминает *Space Invaders* (*Космических завоевателей*). Пусть ваш сын или племянник думает, что получает свою ежедневную порцию разрушительных компьютерных игр, а заодно он будет усовершенствовать свои навыки печати – на всех наших подопытных это сработало.

Пожалуй, самая сюрреалистическая опция – это когда в «Comet Zap» вы выбираете названия растений и с шумом стреляете лазером, набивая слова «маргаритка» и «бегония». Если вам удастся составить такой же абсурдный список слов, пожалуйста, пришлите его нам в редакцию, и мы по возможности включим его в будущий **LXFDVD**.

TuxType вливает в довольно скучное занятие столь желанную дозу веселья – ежедневные десятиминутные занятия с *TuxType* очень быстро улучшат навыки печати у ваших детей.



› Выберите в меню опцию Comet, и лазерные и звуковые эффекты перенесут вас в новый мир: *Космические завоеватели: Гуру Печати.*



Календарь

When

Версия 1.1.12 **Сайт** www.lightandmatter.com/when/when.html

When – календарное приложение. Да, так оно и называется – «*When*» [англ. «когда»]. *When* – это не крутой органайзер и не соперник Google Calendar; это простая и легкая в установке утилита командной строки Unix, которая суммирует все то, что вы должны сделать в определенный день.

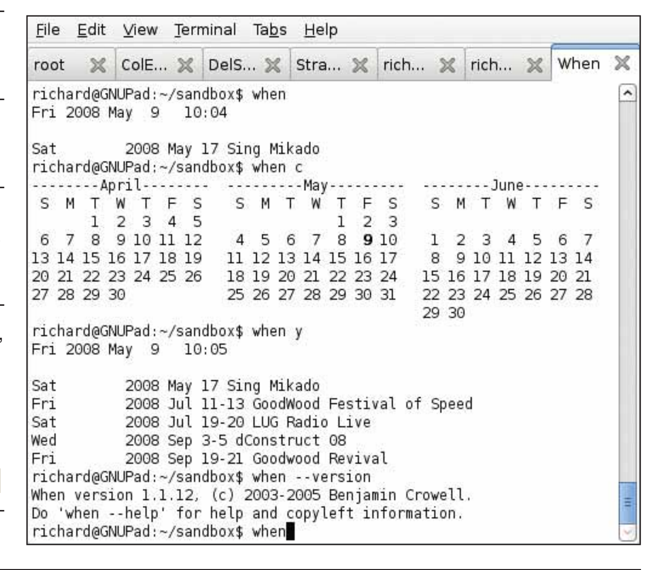
Чтобы запустить ее, наберите when e для установки предпочтений, например, текстового редактора по умолчанию. При этом также создается директория **~/.when** с файлами для ваших настроек и календарной информацией. Добавляйте каждую новую календарную запись в конец файла **calendar**, и *When* будет выводить их все по порядку. По умолчанию он содержит список на две недели, но через различные опции вы можете настроить его так, чтобы он предоставлял списки дел на месяц или год (*см. экранный снимок справа*).

Можно добавлять постоянные мероприятия:

```
w=Tue,7.30PM Репетиция хора
```

а также дни рождения и юбилеи. Если вы добавите *When* в файл конфигурации **.xsession** в директории **home**, вы будете получать напоминание при каждом входе в систему.

Свободные программы, взрослея, иногда становятся слишком перегруженными с точки зрения тех, чьи запросы просты – к счастью, *When* к таковым не относится.



Хранитель файлов

DelSafe

Версия 0.5.0 **Сайт** http://homepage.esoterica.pt/~nx0yew/delsafe

Случайно удалили важный файл? Разделяем вашу боль – но с нами подобного не случается. О нет, мы не идеальны, просто мы установили *DelSafe*.

DelSafe перемещает удаленные файлы в каталог **.Trash** на жестком диске, и простая команда undel вернет их на место. Или, более научно, *DelSafe* подменяет библиотечные функции unlink, rename, open и fopen, используя механизм Linux LD_PRELOAD. Так как в результате происходит просто переименование файлов, он изумительно быстр.

Чтобы управлять данным процессом, в каждой точке монтирования создается файл **.Trash**. Помимо использования команды undel и опций для восстановления более старых версий из **.Trash**, вы можете воскрешать их посредством обозревателя, который позволяет сделать простое фоновое копирование, подобно *Midnight Commander*.

Для установки *DelSafe* сначала запустите **./delsafe-0.5.0.sh**, ответьте на вопросы о расположении файлов и исключениях из защиты *DelSafe*, затем запустите скрипт *Install* и создайте папки **.Trash**:

```
./Install
delsafe_create -c
```

Проверьте, как он работает:

```
touch asterix.gaul
rm asterix.gaul
undel asterix.gaul
```

Вам придется периодически сжимать или чистить мусор. А в остальное время можете расслабиться и даже разрешить другим членам семьи воспользоваться вашим компьютером – вы ведь знаете, что ваши файлы никуда не денутся. LXF



› Delsafe: защита от дурака на вашей Linux-машине.

Также вышли...

Новые и обновленные программы, достойные вашего внимания...

» Mutt Folder List 1.5.17-20080412
Заплатка для почтового клиента *Mutt*, добавляющая боковую панель с отображением всех почтовых папок. www.lunar-linux.org/index.php?page=mutt-sidebar

» Dada Mail 3.0 beta 1
Интуитивная, основанная на web система управления списками рассылки. http://mojo.skazat.com

» ClearHealth 0.1
Система управления медицинской практикой с поддержкой iPhone. www.clear-health.com

» Parapin 1.5.1
Хорошо документированная библиотека C для простого программирования параллельных портов. http://parapin.sourceforge.net

» Kradview 1.0.0
Программа для просмотра Dicom (Digital Imaging and Communications in Medicine – цифровые изображения и коммуникации в медицине). www.orcero.org/irbis/kradview

» Votorola 0.1.2
Программа для подсчета голосов на выборах. http://zelea.com/project/votorola/home.xht

» PLPTools 1.0.3
Соединяет ваш все еще полезный органайзер Psion с вашей Linux-машиной. http://plptools.sourceforge.net

» Ah3 beta9
Система счетов и инвентаризации для автомобильных и прочих запчастей. www.linuxlots.com/~ah3

» 3rd Edition Dungeons & Dragons Character Generator 4.1
Создает новых персонажей прямо в вашем браузере. http://dnd3rd.sourceforge.net

» OpenCOBOL 1.1-pre
Переводит COBOL в C и компилирует полученный код, используя GCC. www.opencobol.org

» Desmume 0.8
Портативный эмулятор DS, способный воспроизводить и демо домашнего изготовления, и коммерческие игры. http://desmume.org

» PDF Split and Merge 1.0.0 beta 3
Делает именно то, что обещано в названии: разделяет на части и объединяет PDF-файлы или их куски. www.pdfsam.org



Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия
ПИ № ФС77-21973 от 14 сентября 2005 года
Выходит ежемесячно. Тираж 6000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ:
ГЛАВНЫЙ РЕДАКТОР
Валентин Синицын info@linuxformat.ru
ВЫПУСКАЮЩИЙ РЕДАКТОР
Родион Водейко
Литературные редакторы
Елена Толстякова
Переводчики
Илья Аввакумов, Александр Бикмеев, Светлана Кривошеина, Александр Казанцев, Алексей Опарин, Валентин Развозжаев
Редактор диска
Александр Кузьменков
Допечатная подготовка
Мария Пучкова, Родион Водейко, Семен Фрумкин
Креативный директор
Станислав Медведев
Технический директор
Денис Филиппов
Директор по рекламе
Денис Игнатов +7 812 965 7236 advert@linuxformat.ru
Генеральный директор
Павел Фролов
Заместитель генерального директора
Софья Виниченко
УЧРЕДИТЕЛИ
частные лица
ИЗДАТЕЛИ
Станислав Медведев, Виктор Федосеев, Павел Фролов

Отпечатано в типографии «Текст», ООО «ППК «Текст»
188680, Ленинградская область, Всеволожский район, Колтуши, д.32
Заказ _____

Пре–пресс: *d.r.i.v.e-group*

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ:
Редактор Пол Хадсон [Paul Hudson] paul.hudson@futurenet.co.uk
Редактор обзоров Грэм Моррисон [Graham Morrison]
Редактор новостей Майк Сондерс [Mike Saunders] mike.saunders@futurenet.co.uk graham.morrison@futurenet.co.uk
Художественный редактор Эфрейн Эрнандес-Мендоса (Efrain Hernandez-Mendoza) efrain.hernandez-mendoza@futurenet.co.uk
Литературные редакторы Эндрью Грегори [Andrew Gregory] agregory@futurenet.co.uk, Мэтт Нейлон [Matt Nailon] mnailon@futurenet.co.uk

Подготовка материалов
Нейл Ботвик [Neil Bothwick], Джон Брэндон [John Brandon], д-р Крис Браун [Dr. Chris Brown], Энди Ченнел [Andy Channelle], Дэн Фрост [Dan Frost], Майкл Дж. Хэммел [Michael J. Hammel], Дэниел Джеймс [Daniel James], Джульетта Кемп [Juliet Kemp], Дейл Стрикленд-Кларк [Dale Strickland-Clark], Ричард Смедли [Richard Smedley], Ник Вейч [Nick Veitch], Алекс Янг [Alex Young], Евгений Балдин, Александр Бикмеев, Андрей Боровский, Владимир Житомирский, Андрей Кузьменко, Алексей Маслий, Галина Пожарина, Алексей Федорчук, Светлана Шапошникова, Павел Фролов, Илья Шпаньков, Никита Шультайс
Художественные ассистенты: Ник Кокс [Nick Cox], Стейси Блек [Stacey Black], Эд Грин [Ed Green], Фил Хэйкрофт [Phil Haycraft]
Ассистент по выпуску: Сара Эдвардс [Sarah Edwardes]
Иллюстрации: Крис Винн [Chris Winn], iStock Photo

КОНТАКТНАЯ ИНФОРМАЦИЯ
UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW
Tel 01225 442244 Email: linuxformat@futurenet.co.uk
РОССИЯ:
Санкт-Петербург (редакция): Лиговский пр. 50 корп. 15, тел. +7 (812) 640-49-90
Представительство в Москве:
м. Авиамоторная, Красноказарменная улица, дом 17. В помещении АТС МЭИ.
Тел/факс: +7(499) 271-49-54.
По вопросам сотрудничества, партнерства, оптовых закупок partner@linuxcenter.ru



в сентябре

SUSE снова на коне?

SUSE, правившая некогда бал в мире Linux, сдала позиции под натиском Ubuntu – но версия 11.0 может изменить баланс сил.

Сообщения об ошибках по-русски

Говорите с Linux на одном языке, диагностируйте типовые проблемы и исправляйте их!

Музыка для наших ушей

Как школы всего мира могут сэкономить с Linux.

Конкуренты Matlab

Вы просили – мы сделали: читайте Сравнение численных пакетов для Linux.

Содержание последующих выпусков может меняться без уведомления